

**PERBAIKAN KUALITAS CITRA DIGITAL
MENGUNAKAN METODE *MULTISCALE
RETINEX* DAN *MEDIAN FILTER***

SKRIPSI

RIZQI HIDAYAT TANJUNG

0701163152



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2020**

**PERBAIKAN KUALITAS CITRA DIGITAL
MENGUNAKAN METODE *MULTISCALE
RETINEX* DAN *MEDIAN FILTER***

SKRIPSI

Diajukan Untuk Memenuhi Syarat Mencapai Gelar Sarjana Komputer

RIZQI HIDAYAT TANJUNG

0701163152



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2020**

PERSETUJUAN SKRIPSI

Hal : Surat Persetujuan Skripsi
Lamp : -
Kepada Yth.,
Dekan Fakultas Sains dan Teknologi
Universitas Islam Negeri Sumatera Utara Medan

Assalamu'alaikum Wr. Wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengatakan perbaikan, maka kami selaku pembimbing berpendapat bahwa skripsi saudara,

Nama : Rizqi Hidayat Tanjung
Nomor Induk Mahasiswa : 070116312
Program Studi : Ilmu Komputer
Judul : Perbaikan Kualitas Citra Digital Menggunakan
Metode *Multiscale Retinex* Dan *Median Filter*

Dapat disetujui untuk segera dimunaqasyahkan. Atas perhatiannya kami ucapkan terimakasih.

Medan, 22 Oktober 2020

05 Rabi'ul-Awal 1442

Komisi Pembimbing,

Pembimbing Skripsi I,



Dr. Mhd Furqan, S.Si., M.Comp.Sc
NIP. 198008062006041003

Pembimbing Skripsi II,



Abdul Halim Hasugian, M.Kom.
NIB.1100000113

SURAT PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini,

Nama : Rizqi Hidayat Tanjung
Nomor Induk Mahasiswa : 0701163152
Program Studi : Ilmu Komputer
Judul : Perbaikan Kualitas Citra Digital Menggunakan
Metode *Multiscale Retinex* Dan *Median Filter*

Menyatakan bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya. Apabila di kemudian hari ditemukan plagiat dalam skripsi ini maka saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi lainnya sesuai dengan peraturan yang berlaku.

Medan, 14 Oktober 2020



Rizqi Hidayat Tanjung
NIM. 0701163152

ABSTRAK

Pada saat ini citra digital banyak sekali digunakan oleh orang-orang untuk mengabadikan momen ataupun hal-hal penting lainnya, citra digital itu sendiri merupakan hasil dari pengambilan gambar dengan kamera digital, walaupun kamera digital saat ini sudah dilengkapi fitur yang mendukung hasil gambar, namun tidak semua citra digital yang dihasilkan sesuai dengan keinginan kita, itu terjadi karena adanya masalah pada kualitas citra, situasi dan kondisi pada saat proses pemotretan citra akan mempengaruhi pada kualitas citra yang menyebabkan citra hasil menjadi terang atau gelap. Maka dari itu dibutuhkan perbaikan kualitas citra agar citra yang dihasilkan sesuai dengan keinginan kita, pada penelitian ini metode yang dipakai dalam perbaikan kualitas citra yaitu menggunakan metode *Multiscale Retinex* dan *Median Filter*. Proses dari *Multiscale Retinex* akan menghasilkan citra yang lebih terang dibandingkan dengan citra asli yang minim intensitas cahaya, sedangkan proses *Median filter* akan menghasilkan citra yang lebih jelas karena dapat mengurangi *derau/noise* pada citra, parameter yang dipakai pada penelitian ini adalah *Histogram*, dari beberapa data yang telah diteliti *histogram* menunjukkan grafik nilai intensitas *pixel* rata-rata mendekati nol (0) setelah diproses dengan metode *Multiscale Retinex* dan *Median Fiter* nilai intensitas *pixel* rata-rata menunjukkan perubahan grafik mendekati 250 yang menunjukkan peningkatan kecerahan pada citra digital yang semakin terang.

Kata Kunci : Citra Digital, *Multiscale Retinex*, *Median Filter*, *Histogram*

ABSTRACT

At this time digital images are widely used by people to capture moments or other important things, the digital image itself is the result of taking pictures with a digital camera, even though today's digital cameras are equipped with features that support image results, but not All digital images are produced according to our wishes, it occurs because of problems with the image quality, the situation and conditions during the image shooting process will affect the image quality which causes the resulting image to be light or dark. Therefore, it is necessary to improve the image quality so that the resulting image is in accordance with our wishes, in this study the method used in image quality improvement is using the Multiscale Retinex and Median Filter methods. The process of Multiscale Retinex will produce a brighter image compared to the original image with minimal light intensity, while the Median filter process will produce a clearer image because it can reduce noise in the image, the parameter used in this study is the Histogram, from some data. which has been examined the histogram shows the graph of the average pixel intensity value close to zero (0) after being processed with the Multiscale Retinex and Median Fiter method, the average pixel intensity value shows a graph change approaching 250 which indicates an increase in brightness in a brighter digital image.

Keywords: *Digital Image, Multiscale Retinex, Median Filter, Histogram*

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh.

Alhamdulillahirobbil'alamin segala puji dan syukur penulis sampaikan kepada Allah Subhana Wata'ala, karena dengan rahmat-Nya penulis mampu menyelesaikan skripsi ini. Shalawat beserta salam semoga senantiasa tercurah Kepada Nabi Muhammad Shalallahu Alihi Wassalam.

Penulisan skripsi ini bertujuan sebagai persyaratan untuk memperoleh gelar sarjana program studi Ilmu Komputer Fakultas Sains dan Teknologi, Universitas Islam Negeri Sumatera Utara. Judul yang diteliti oleh penulis ialah "Perbaikan Kualitas Citra Digital Menggunakan Metode *Multiscale Retinex* dan *Median Filter*".

Skripsi ini diberikan untuk orang-orang yang sangat berjasa dalam hidup penulis yaitu Ayah tercinta Drs. Marwan Tanjung, dan Ibu tersayang Dra. Neni Triani, dan Abang, kakak dan Adik saya, Fazlurrahman Wahid Tanjung, Rizqi Yatussaniyah Tanjung dan Rizki Ramadhan Tanjung yang telah menjadi keluarga terbaik, yang selalu memberi semangat, motivasi, arahan dan doa yang tulus yang tentu sangat membantu penulis dalam pengerjaan skripsi ini. Semoga Allah selalu memberikan kesehatan, karunia dan hidayah-Nya, terima kasih ayah dan Ibu yang sangat luar biasa dalam memdidik penulis, yang setiap harinya menghubungi saya untuk memotivasi saya, mungkin hanya ucapan terima kasih inilah yang bisa penulis persembahkan untuk saat ini, Semoga berkat doa-doa ayah dan ibu saya bisa menjadi orang berguna bagi masyarakat dan negara,

Penulis sadar pada tahap proses penyusunan skripsi ini sangat lah sulit, berupa kendala maupun cobaan selalu ada. Akan tetapi karena ketekunan dan pantang menyerah, usaha dan doa, serta bimbingan dari berbagai pihak, kendala dan cobaan yang muncul dapat terselesaikan. Maka dari itu, penulis ingin memberikan ucapan terimakasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Saidurrahman, M. Ag selaku Rektor Universitas Islam Negeri Sumatera Utara Medan.
 2. Bapak Dr. H.M. Jamil, MA selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara Medan.
 3. Bapak Dr. Mhd. Furqan, S.Si.,M.Comp.Sc selaku Ketua Program Studi Ilmu Komputer, Universitas Islam Negeri Sumatera Utara Medan, sekaligus sebagai pembimbing I.
 4. Bapak Ilka Zufria, M.Kom selaku Sekretaris Prodi Ilmu Komputer, Universitas Islam Negeri Sumatera Utara Medan.
 5. Bapak Abdul Halim Hasugian, M.Kom sebagai pembimbing skripsi II saya.
 6. Bapak Heri Santoso, M.Kom sebagai dosen pembimbing akademik.
 7. Seluruh staf pengajar dan pegawai lingkungan Prodi Ilmu Komputer Universitas Islam Negeri Sumatera Utara.
 8. Teman-teman seperjuangan saya ilmu komputer angkatan 2016 terkhusus teman-teman kelas saya Ilmu Komputer 3 yang sangat saya sayangi dan rindukan
 9. Serta semua pihak yang turut membantu penulis dalam menyelesaikan skripsi ini yang tidak dapat penulis sebutkan satu-persatu.
- Semoga Skripsi ini bermanfaat bagi pembaca.

Medan, 22 Oktober 2020

Penyusun

Rizqi Hidayat Tanjung

DAFTAR ISI

PERSETUJUAN SKRIPSI

SURAT PERNYATAAN KEASLIAN SKRIPSI

ABSTRAK i

ABSTRACT ii

KATA PENGANTAR..... iii

DAFTAR ISI.....v

DAFTAR GAMBAR..... vii

DAFTAR TABEL viii

DAFTAR LAMPIRAN ix

BAB I PENDAHULUAN.....1

1.1 Latar Belakang1

1.2 Rumusan Masalah.....3

1.3 Batasan Masalah4

1.4 Tujuan Penelitian4

1.5 Manfaat Penelitian4

BAB II TINJAUAN PUSTAKA.....5

2.1 Pengertian Citra5

2.1.1 Citra Analog6

2.1.2 Citra Digital6

2.2 Pengolahan Citra.....8

2.3 Perbaikan Kualitas Citra10

2.4 *Retinex*12

2.4.1 *Singlescale Retinex*.....12

2.4.2 *Multiscale Retinex*.....13

2.5 *Median Filter*15

2.6 *Histogram*18

2.7 *Matlab*.....18

2.8 *Flowchart*.....19

2.9 Penelitian Terkait.....	21
BAB III METODOLOGI PENELITIAN	22
3.1 Tempat dan Waktu Penelitian.....	22
3.1.1. Tempat Penelitan	22
3.1.2. Waktu & Jadwal Pelaksanaan Penelitian.....	22
3.2 Bahan dan Alat Penelitian	22
3.2.1. Perangkat Keras	22
3.2.2. Perangkat Lunak	22
3.3 Cara Kerja.....	22
3.3.1 Perencanaan	22
3.3.2 Teknik Pengumpulan Data	23
3.3.3 Analisa Kebutuhan	24
3.3.4 Perancangan.....	25
3.3.5 Pengujian	31
3.3.6 Penerapan/Penggunaan	31
BAB IV HASIL DAN PEMBAHASAN	32
4.1 Pembahasan	32
4.1.1 Analisis Data	32
4.1.2 Representasi Data	32
4.1.3 Hasil Analisis Data	61
4.1.4 Perancangan.....	61
4.2 Hasil	63
4.2.1 Pengujian	63
4.2.2 Penerapan	73
BAB V KESIMPULAN DAN SARAN	74
5.1 Kesimpulan	74
5.2 Saran	74
DAFTAR PUSTAKA	75
LAMPIRAN -LAMPIRAN	

DAFTAR GAMBAR

Gambar	Judul Gambar	Halaman
2.1	Koordinat Citra Digital.....	7
2.2	Matriks f	13
2.3	Kernel g	13
2.4	Citra Sample	16
2.5	Citra Asli	17
2.6	Hasil Citra Output	17
3.1	Tahap-tahap Perencanaan.....	23
3.2	<i>Flowchart</i> Sistem	25
3.3	<i>Flowchart Multiscale Retinex</i>	26
3.4	<i>Flowchart Median Filter</i>	27
3.5	Rancangan Dialog	28
3.6	Form <i>Histogram</i>	30
4.1	Citra Asli	33
4.2	Potongan Nilai <i>Pixel</i> Citra RGB	33
4.3	<i>Flowchart</i> Sistem Aplikasi.....	62
4.4	Tampilan Antar Muka	63
4.5	Tampilan Awal Aplikasi	64
4.6	Penginputan Citra.....	64
4.7	Hasil Proses MSR.....	65
4.8	Hasil Proses <i>Median Filter</i>	65
4.9	Simpan Citra.....	66
4.10	<i>Histogram</i>	66

DAFTAR TABEL

Tabel	Judul Tabel	Halaman
2.1	Citra Input	16
2.2	Proses pencarian kernel	16
2.3	Hasil Output 5x5 pixel	17
2.4	Simbol-simbol <i>Flowchart</i>	20
4.1	Hasil Perhitungan Konvolusi SSR_1	48
4.2	Hasil SSR_1	49
4.3	Hasil Perhitungan Konvolusi SSR_2	51
4.4	Hasil SSR_2	52
4.5	Hasil Perhitungan Konvolusi SSR_3	54
4.6	Hasil SSR_3	56
4.7	Hasil Perhitungan <i>Multiscale Retinex</i>	57
4.8	Hasil Normalisasi	57
4.9	Nilai <i>Pixel Multiscale Retinex</i>	58
4.10	Hasil Output <i>Median Filter</i>	61
4.11	Hasil Citra Digital dan <i>Histogram</i>	67

DAFTAR LAMPIRAN

Lampiran	Judul Lampiran
1	Tools Matlab Gui untuk Memperbaiki Kualitas citra digital menggunakan metode <i>Multiscale Retinex</i> dan <i>Median Filter</i>
2	Kode Program Matlab untuk Memperbaiki Kualitas Citra Digital Menggunakan Metode <i>Multiscale Retinex</i> dan <i>Median Filter</i>
3	Data Uji
4	Kartu Bimbingan
5	Daftar Riwayat Hidup

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada zaman sekarang ini, pengaruh dari bidang teknologi terhadap kehidupan manusia sangatlah penting. Hal ini dikarenakan seseorang selalu menggunakan pemikirannya dalam menyelesaikan berbagai masalah yang dihadapinya dalam kehidupan. Seiring dengan semakin majunya perkembangan teknologi, kemajuan dalam hal ilmu pengetahuan juga turut berperan sehingga tidak dapat dihindari melihat kenyataan ini manusia dituntut untuk memanfaatkan dengan baik teknologi dan ilmu pengetahuan tersebut (Ngafifi, 2014).

Teknologi dan ilmu pengetahuan termasuk dalam sarana peningkatan kadar keimanan makhluk terhadap Sang Pencipta Allah SWT. Kekuasaan Allah SWT akan lebih jelas terhadap manusia yang memiliki ilmu pengetahuan jika dibandingkan dengan manusia yang kurang memuntut ilmu pengetahuan (Fakhri, 2010).

Allah SWT berfirman:

يٰمَعْشَرَ الْجِنِّ وَالْإِنسِ إِنِ اسْتَطَعْتُمْ أَنْ تَنْفُذُوا مِنْ أَقْطَارِ السَّمٰوٰتِ وَالْأَرْضِ فَانْفُذُوْا
لَا تَنْفُذُوْنَ اِلَّا بِسُلْطٰنٍ ۝

Artinya: “Hai jama'ah jin dan manusia, jika kamu sanggup menembus (melintasi) penjuru langit dan bumi, maka lintasilah, kamu tidak dapat menembusnya kecuali dengan kekuatan. (QS. Ar Rahman: 33)

Menurut firman Allah SWT tersebut diketahui bahwa sebagai makhluk ciptaan-Nya kita dituntut untuk menuntut ilmu pengetahuan dan teknologi sebagai suatu bekal dalam menjalankan kehidupan di dunia. Dengan memiliki ilmu pengetahuan dan teknologi sebagai bekal diharapkan manusia mampu melewati perkembangan zaman sesuai dengan apa yang diharapkan.

Pada dasarnya teknologi yang dibangun diharapkan mampu memberikan kemudahan kepada manusia dalam menjalani kehidupan sehingga dapat dimanfaatkan dengan baik dalam membantu manusia menjalani aktivitas.

Perkembangan teknologi yang sangat pesat menandakan bahwa dalam bidang industri teknologi tingkat dunia kini telah menunjukkan kearah yang termasuk kedalam tingkat mobilitas kompleks (Muhajir Affandi, 2018).

Salah satu bentuk kemajuan di bidang teknologi adalah adanya suatu penggambaran berupa ilustrasi dari objek bersifat *optic* misalnya saja berupa foto, bersifat analog yang berupa sinyal-sinyal video seperti gambar pada layar televisi, atau bersifat *digital* yang mampu untuk langsung disimpan pada suatu media penyimpanan yang disebut dengan citra (Supriadi et al., 2017).

Namun terkadang tidak selamanya citra yang dihasilkan sesuai dengan keinginan. Citra terkadang mengalami gangguan berupa degradasi atau penurunan mutu hal ini dikarenakan adanya gangguan *derau/noise*, kontras terlalu tinggi, buram dan lain sebagainya. Adapun gangguan ini disebabkan oleh berbagai aktor berupa minimnya sinar cahaya ketika proses mengambil gambar, minimnya resolusi *pixel* dari kamera yang dipakai, dan terbatasnya kemampuan dalam pengambilan gambar (Syamsuddin, 2019).

Proses menangani gambar yang terdegradasi dan meningkatkan kualitasnya dikenal sebagai peningkatan gambar (Parihar & Singh, 2019). Peningkatan gambar penting untuk analisis gambar, diagnosis, dan tampilan (Li et al., 2011). Tujuan peningkatan gambar adalah untuk mendapatkan detail gambar yang lebih halus dan menyoroti informasi yang berguna (Bangalore, n.d.).

Maka untuk memberikan solusi terhadap masalah tersebut diperlukan adanya suatu sistem pengolahan citra berdasarkan metode yang digunakan dalam memperbaiki masalah tersebut. Metode yang digunakan adalah *Multiscale Retinex*. *Multiscale Retinex* merupakan algoritma peningkatan gambar nonlinear yang mensimulasikan sistem visual manusia. Algoritma *retinex* memiliki konstansi warna, rentang dinamis tinggi dan juga dapat mempertajam detail (Hong-wei & Jiu-gen, 2018). Dengan digunakannya metode *Multiscale Retinex* diharapkan mampu untuk mengatasi masalah kualitas citra pada bagian

pencerahannya nantinya diperoleh hasil citra yang sesuai dengan keinginan. Menurut Edwin Land pada tahun 1971 mengenai Metode *retinex*. Algoritma *retinex* memiliki tujuan yang utama yaitu untuk membedakan citra *S* menjadi dua *image* yang berbeda, seperti *reflectance image* *R* dan *illumination image* *L* pada setiap titik (x,y) di *image* domain (Supriadi et al., 2017).

Selain *Retinex*, *Median Filter* juga merupakan salah satu metode dalam penelitian yang mampu untuk memperbaiki kualitas citra digital. *Median filter* termasuk kedalam salah satu cara dari peningkatan kualitas citra dalam domain spasial. Cara ini tergolong dalam kategori *non linear filtering*, yang berarti bahwa nilai *pixel output* yang dibuat oleh median dari lingkungan *mask* yang ditentukan. Cara menentukan median dengan mengurutkan nilai dari *pixel mask* yang telah ditentukan, lalu mencari nilai tengahnya

Metode *median filter* termasuk dalam filter non linear yang dikaji lebih dalam oleh Tukey. Metode median filter memiliki fungsi untuk melakukan pengurangan terhadap noise dan memberikan penghalusan terhadap citra. Dikatakan non linear dikarenakan cara kerja penapis ini tidak tergolong kedalam golongan operasi *konvolusi*, operasi *non linear* dihitung dengan cara mengurutkan nilai intensitas sekelompok *pixel* yang diproses dengan nilai tertentu (Maulana et al., 2016).

Berdasarkan latar belakang tersebut, maka penulis berinisiatif untuk melakukan suatu penelitian terhadap citra digital agar hasil citra yang dihasilkan lebih bagus dan terlihat jelas setelah dilakukan perbaikan kualitas citra. Sehingga dilakukanlah sebuah penelitian yang berjudul **Perbaikan Kualitas Citra Digital dengan Menggunakan Metode *Multiscale Retinex* dan *Median Filter***.

1.2 Rumusan Masalah

Berdasarkan penjelasan latar belakang masalah di atas, maka rumusan masalah yang didapat penulis yaitu :

1. Bagaimana melakukan analisis metode *Multiscale Retinex* dan *Median Filter* dalam Perbaikan kualitas citra digital ?

2. Bagaimana menghasilkan sistem aplikasi yang mampu melakukan perbaikan kualitas citra digital dengan metode *Multiscale Retinex* dan *Median Filter* ?

1.3 Batasan Masalah

Penelitian difokuskan pada beberapa hal berikut ini :

1. Citra yang diteliti hanya citra digital yang minim intensitas cahaya.
2. Data citra yang diproses format jpg.
3. Metode *Multiscale Retinex* hanya dipakai pada proses peningkatan kecerahan citra digital.
4. Penggunaan metode *Median Filter* dipakai pada proses pengurangan derau pada citra digital.
5. Perangkat lunak yang digunakan adalah Matlab 2015.
6. Metode yang digunakan *Multiscale Retinex* dan *Median Filter*.

1.4 Tujuan Penelitian

Adapun tujuan yang ingin dicapai ialah :

1. Untuk melakukan analisis metode *Multiscale Retinex* dan *Median Filter* dalam Perbaikan kualitas citra digital.
2. Untuk merancang sistem aplikasi yang mampu melakukan perbaikan kualitas citra digital dengan metode *Multiscale Retinex* dan *Median Filter*.

1.5 Manfaat Penelitian

Adapun manfaat yang diperoleh dari penelitian ini ialah dapat memperjelas dan mencerahkan citra digital, yang awalnya kualitas citra tersebut kurang baik menjadi lebih baik.

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Citra

Citra yaitu suatu komponen yang memiliki peranan begitu penting sebagai suatu bentuk yang berasal dari informasi visual (Furqan & Siregar, 2020). Sedangkan Citra menurut Kamus Webster yaitu suatu kemiripan, gambaran, dari objek benda, misalnya suatu hasil gambar milik seseorang yang berasal dari kamera orang tersebut, foto sinar X-thorax yang diambil untuk mewakili gambar dari bagian tubuh seseorang dan lain-lain (Tarigan et al., 2016).

Citra merupakan gambaran dari sesuatu hal berupa objek. Citra adalah output sistem dalam perekaman data dan bersifat optic berupa foto yang bersifat analog ataupun sinyal video seperti gambar pada layar televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan (Supriadi et al., 2017).

Suatu citra terbagi menjadi 2 bagian, citra diam (*still image*) dan citra bergerak (*moving image*). Pada citra diam dihasilkan suatu citra yang tunggal dan tidak ada pergerakan. Sedangkan pada citra bergerak terdapat kumpulan citra diam yang ditayangkan secara berurutan (*sequential*) sehingga memberi gambaran pada mata seperti seolah gambar dapat bergerak (Gunadarma, 2005).

Citra termasuk dalam bentukan informasi yang dibutuhkan seseorang selain tulisan, video, ataupun audio. Informasi berupa citra bukan hanya untuk berkomunikasi dengan sesama manusia melainkan manusia dengan mesin. Informasi yang disampaikan citra dapat berbeda tanggapan yang disampaikan antara manusia yang satu dengan lainnya. Maksudnya yaitu, informasi dalam sebuah citra bersifat subjektif tergantung pemikiran dan kebutuhan perseorangan. Maka dari itu diperlukan suatu pengolahan citra sehingga diperoleh citra yang sesuai dengan keinginan (Sulistiyanti, 2016).

Jenis citra dapat dibedakan menjadi dua jenis antara lain, citra analog & citra digital. Citra analog adalah suatu citra dengan sifat kontinu yang asalnya muncul dari suatu sistem optik pada saat menerima sinyal analog. Sedangkan citra digital yaitu citra yang diperoleh melalui proses digitalisasi terhadap citra analog. Citra digital dibagi menjadi tiga jenis yaitu citra warna, citra grayscale dan citra biner (Tarigan et al., 2016).

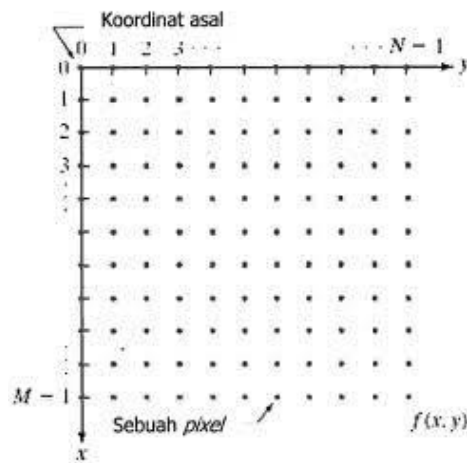
2.1.1 Citra Analog

Citra analog adalah citra yang bersifat berkelanjutan, seperti gambar pada monitor televisi, lukisan, foto sinar-X, foto yang tercetak dikertas foto pemandangan, hasil *CT scan*, gambar-gambar yang terekam pada pita kaset, dan lain-lain merupakan contoh citra analog. Citra analog tidak dapat diproses dikomputer secara langsung dan tidak dapat direpresentasikan didalam komputer. Oleh karena itu agar citra dapat diproses oleh komputer, maka dilakukan proses konversi citra analog ke citra digital (Sutoyo, 2009).

2.1.2 Citra Digital

Secara umum, citra digital mengarah kepada suatu pengolahan gambar 2 dimensi menggunakan sebuah komputer. Pada kaitan yang lebih meluas, pengolahan citra digital mengarah pada setiap proses data 2 dimensi. Citra digital ialah suatu larik (*array*) yang mengandung suatu nilai nyata ataupun kompleks yang direpresentasikan beserta urutan bit tertentu.

Citra didefinisikan dengan fungsi $f(x,y)$ berukuran N baris dan M kolom, dengan x dan y yaitu koordinat spasial, dan amplitud f di titik koordinat (x,y) disebut intensitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai x,y, dan nilai amplitud f secara menyeluruh berhingga (*finite*) dan bernilai diskrit maka dapat disebut bahwa citra itu adalah citra digital (Darma Putra, 2010).



Gambar 2.1 Koordinat Citra Digital

Citra digital juga dapat ditulis dalam bentuk matriks sebagai berikut:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix}$$

Nilai pada suatu irisan antara baris dan kolom (pada posisi x,y) disebut dengan picture elements, image elements, pels, atau *pixel*. Istilah terakhir (*pixel*) paling sering digunakan pada citra digital.

Pada Pengolahan Citra digital pada umumnya, citra digital dapat dibagi menjadi 3 jenis yaitu: (Kusumanto, 2011)

1. Citra Warna

Pada bagian ini, masing-masing *pixel* memiliki berbagai warna tersendiri, warna tersebut adalah hijau (green), merah (Red), dan biru (blue). Apabilasetiap warna memilki range 0-255, maka jumlahnya adalah $255^3 = 16.581.375$ (16 K) variasi warna berbeda pada gambar. Citra warna ini terdiri dari tiga matriks yang mewakili nilai-nilai hijau, merah, dan biru untuk setiap *pixel* nya.

2. Citra *Grayscale*

Pada setiap *pixel* Citra *grayscale* memiliki jenis warna bergradasi dari putih sampai hitam. Ini mengartikan bahwa pada setiap *pixel* diwakili oleh 8 bit, atau 1 byte, kadar warna pada citra *grayscale* sangat tepat ketika dimanfaatkan untuk pengolahan file gambar. Salah satu bentuk pemanfaatannya pada bagian kedokteran yaitu *X-ray*.

3. Citra Biner

Pada setiap *pixel* citra biner hanya berasal dari warna hitam dan putih, hal ini dikarenakan hanya memiliki dua warna pada setiap *pixel* (0 dan 1) atau ketika dalam 8 bit (0 dan 255), sehingga efektif digunakan dalam penyimpanan.

2.2 Pengolahan Citra

Pengolahan dalam KBBI adalah suatu kegiatan yang terdiri dari proses dalam mengusahakan sesuatu sehingga dapat menjadi sesuatu yang lebih baik. Maka citra menurut KBBI yaitu gambaran atau rupa yang dalam hal ini gambar tersebut diperoleh dari sistem visual. Secara lebih menyeluruh pengolahan citra merupakan upaya untuk menghasilkan suatu citra menjadi lebih baik lagi. Dengan kata lain pengolahan citra dilakukan untuk melakukan suatu proses dalam memasukkan citra sehingga diperoleh hasil yang sesuai dengan keinginan.

Awal mulanya pengolahan citra digital hanya sebatas penggunaan untuk mengubah citra analog menjadi citra digital serta memperbaiki kualitas citra. Namun perkembangan waktu menjadikan peralatan menjadi pendukung pengolahan citra menjadi lebih beragam. Dengan suatu algoritma pengolahan citra, sensor penglihatan buatan berupa kamera diharapkan dapat menggantikan fungsi sensor penglihatan pada manusia. Kecepatan pemrosesan dalam komputer yang semakin berkembang memungkinkan proses pengolahan citra digital dilakukan secara *realtime*. Maka dari itu, dengan adanya perkembangan memori menandakan bahwa citra analog dapat dikodekan menjadi citra warna digital yang mendekati warna aslinya (Sulistiyanti, 2016).

Pengolahan citra digital merupakan suatu pembelajaran untuk mengetahui cara-cara mengolah citra. Citra disini yaitu gambar yang tidak bergerak atau diam (foto) ataupun gambar gerak (dari webcam). Sedangkan yang dimaksud digital yaitu penolahan gambar dilakukan secara digutal menggunakan perangkat komputer (Kusumanto, 2011)

Proses yang dijalankan pada pengolahan citra dapat dikelompokkan menjadi beberapa jenis yaitu antara lain (Hasugian & Zufira, 2018) :

- a. Perbaikan kualitas citra (*image enhancement*), pada jenis ini citra akan dimanipulasi melalui parameter-parameter citra yang bertujuan untuk memperbaiki kualitas citra. Padajenis ini, ciri-ciri khusus yang ada lebih ditonjolkan. Contoh-Nya adalah perbaikan kontras gelap menjadi terang, penajaman (*sharpening*), perbaikan tepian citra, penapisan derau (*noise filtering*), dan pemberian warna semu (*pseudocoloring*),
- b. Pemugaran citra (*image restoration*), pada jenis ini memiliki tujuan untuk meniadakan atau meminimalisir gangguan pada citra. Tujuan pemugaran citra sama hal-Nya dengan perbaikan citra hanya saja yang membedakan pada pemugaran citra disebabkan karena adanya degradasi gambar. Contoh operasi pemugaran antara lain penghilangan kesamaran (*deblurring*) dan penghilangan derau (*noise*).
- c. Pemampatan citra (*image compression*),digunakan agar citra dapat dipresentasikan dalam bentuk yang lebih kompak, sehingga membutuhkan penyimpanan yang lebih minim. Hal terpenting yang harus diperhatikan dalam pemugaran adalah citra yang sudah dimampatkan harus tetap memiliki gambar yang bagus.
- d. Segmentasi citra (*image segmentation*), adapun tujuan dari proses ini ialah membagi suatu citra kedalam beberapa bagian dengan suatu ciri tertentu. Jenis operasi ini ada kaitan-Nya dengan pengenalan pola.
- e. Pengorakan citra (*image analysis*), betujuan menghitung suatu besaran kuantitatif dari citra agar memperoleh diskripsinya. Teknik pengorakan citra mengekstrasi kriteria tertentu yang membantu dalam mengenal objek. Contoh-Nya deteksi tepi objek .

- f. Rekonstruksi citra (*image reconstruction*), digunakan untuk membentuk kembali objek dari suatu hasil citra proyeksi. Operasi rekonstruksi banyak diaplikasikan dalam bidang medis.

2.3 Perbaikan Kualitas Citra

Sebagian besar citra tidak sebanding dengan yang diinginkan. Itu terjadi dikarenakan beberapa hal antara lain, gangguan yang menutupi objek yang sedang difoto, dan lensa kamera yang berdebu. Oleh karena itu pengolahan citra digital sangat dibutuhkan untuk melakukan proses dalam memperbaiki kualitas citra. Biasanya dalam suatu citra terdapat penambahan tulisan atau lainnya yang perlu dihilangkan atau kita ingin melakukan perbaikan terhadap bagian tertentu dari sebuah citra yang dapat merusak citra, hal ini dapat dilakukan dengan menggunakan *image enhancement*. Perbaikan kualitas yaitu suatu teknik restorasi citra dengan cara mengurangi penurunan kualitas citra yang diawali selama pembentukan citra tersebut (Hasugian & Zufira, 2018).

Perbaikan kualitas citra (*image enhancement*) adalah tahap awal dari pengolahan citra (*image processing*). Perbaikan kualitas diperlukan karena sering ditemukan citra yang memiliki kualitas tidak baik, seperti citra yang memiliki derau (*noise*) pada saat proses pengiriman gambar, citra terang/gelap, kabur, citra kurang tajam, dan lainnya. Pada tahap ini kualitas citra dapat dilakukan perbaikan agar citra dapat dimanfaatkan untuk aplikasi lebih lanjut, misalnya aplikasi pengenalan (*recognition*) objek pada citra (Purba, 2017).

Adapun maksud dari perbaikan kualitas citra yaitu proses untuk menghasilkan citra yang lebih mudah diinterpretasikan terhadap mata manusia. Memproses karakteristik tertentu yang terdapat pada citra lebih diperjelas kemunculannya. Secara matematis, *image enhancement* dapat diterjemahkan sebagai proses mengubah citra $f(x,y)$ menjadi $f'(x,y)$ sehingga karakteristik yang dilihat pada $f(x,y)$ lebih ditonjolkan.

Proses yang ada dalam perbaikan citra (*image enhancement*) misalnya, pengubahan kecerahan citra (*image brightness*), perbaikan kontras,

pengurangan derau, pelembutan citra (*image smoothing*), penajaman citra (*image sharpening*).

2.3.1 Kecerahan Citra (Image Brightness)

Untuk memperoleh citra terang ataupun citra gelap, maka ada proses perubahan intensitas citra yang dilakukan, yang dalam hal ini dikenal sebagai penyesuaian tingkat cerahnya gambar. Kecerahan gambar dapat dilakukan perbaikan yaitu mengurangi ataupun menambahkan nilai dari setiap *pixel* yang ada pada citra. Dampaknya yaitu histogram citra mengalami pergeseran (Munawaroh & Sutanto, 2010).

Secara matematis operasi ini ditulis sebagai berikut :

$$f'(x, y) = f(x, y) + b$$

Jika b positif, kecerahan gambar bertambah, sebaliknya jika b negative kecerahan gambar berkurang.

2.3.2 Derau (Noise)

Merupakan suatu gangguan yang menyebabkan kualitas citra menjadi tidak bagus. Adapun penyebab terjadinya derau pada citra bisa terjadi oleh gangguan fisik (optik) sehingga menciptakan proses pengolahan citra yang tidak sempurna. Contoh derau yaitu adanya bintik hitam ataupun putih yang muncul secara acak kedalam sebuah citra. Sebutan yang dikenal dengan bintik acak ini ialah derau salt & pepper. Penyebab lain noise datang dari menyimpangnya data digital yang dikirimkan sehingga pada saat alat penerima data gambar menerima gambar tersebut bisa mengganggu kualitas citra. Beberapa jenis noise yang sering diproses dalam pengolahan citra yaitu *gaussian noise*, *localvar noise*, dan *salt and pepper noise* (Sholihin et al., 2013).

2.4 Retinex

Metode *retinex* disampaikan oleh Edwind Land pada tahun 1971. Tujuan utama dari algoritma *retinex* adalah memisahkan citra S menjadi 2 (dua) *image* yang berbeda, yaitu *reflectance image* R dan *illumination image* L pada setiap titik (x,y) di *image* domain (Supriadi et al., 2017).

2.4.1 Metode SSR (Single-Scale Retinex)

Singlescale Retinex (SSR) merupakan proses awal sebelum proses *Multiscale Retinex* (MSR), dengan melalui tiga tahap SSR. Banyak jurnal yang menyebutkan bahwa *Singlescale Retinex* (SSR) merupakan *Retinex* skala tunggal karena hanya menggunakan satu fungsi *surround* untuk mencari tahu iluminasi dengan gambar yang berbelit-belit, berikut rumus SSR :

$$R_i(x,y) = \log l_i(x,y) - \log [F(x,y) * l_i(x,y)]$$

Dimana $R_i(x,y)$ adalah Output *Retinex*, $l_i(x,y)$ merupakan distribusi citra pada *pixel* ke (x,y) . Simbol “*” Menyatakan operator konvolusi. Simbol I menjelaskan warna, antara lain Merah(R), Hijau(G), dan Biru(B), $F(x,y)$ menyatakan fungsi Gaussian yang didefinisikan :

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

keterangan, $G(x,y)$ adalah *Gaussian kernel* pada *pixel* (x,y) . Simbol π atau π adalah konstanta yaitu $22/7$. Simbol σ ialah nilai sigma. Simbol (x,y) merupakan koordinat *pixel*. Sedangkan simbol e adalah ketetapan yaitu 2.7182818246 *Gaussian filter* dipakai pada citra agar citra lebih halus dalam memproses suatu gambar. untuk menghilangkan *noise* dan meningkatkan kualitas detail citra *Gaussian filter* juga bisa digunakan.

Konvolusi adalah perhitungan yang dilakukan dengan melakukan perkalian antara matriks f dan kernel g . Metode Konvolusi bisa dinyatakan dalam matriks, yang mana pada setiap elemen matriks penapis merupakan koefisien konvolusi. Hasil dari perkalian konvolusi akan disimpan kedalam matriks yang baru dilakukan dengan cara melakukan pergeseran kernel *pixel* per *pixel*.

Berikut hasil ilustrasi konvolusi, yaitu :

		A1	A2	A3		
		A4	A5	A6		
		A7	A8	A9		

$f(x,y)$

Gambar 2.2 matriks f

A	B	C
D	E	F
G	H	I

$g(x,y)$

Gambar 2.3 kernel g

Maka proses perhitungan *konvolusi* dapat dilakukan dengan : $f(x,y)*g(x,y)$

$$f(x,y) = AA1 + BA + CA + DA + EA + FA + GA + HA + IA$$

2.4.2 Metode *Multiscale Retinex* (MSR)

Multiscale Retinex (MSR) adalah generalisasi dari *retinex* (SSR) skala tunggal, yang, pada gilirannya, didasarkan pada versi terakhir dari *retinex* pusat / surround Land (Rahman et al., 2020). *Multiscale Retinex* adalah generalisasi dari

skala tunggal (SSR), yang memiliki batasan sehingga dikembangkan lah MSR batasannya yaitu :

1. SSR membatasi dari segi kompresi skala minim dan skala besar dimana skala minim membuat kompresi jarak dinamis pada citra dan memungkinkan citra sangat luas dikompresi pada bagian gelap dan melemahkan bagian yang terang. Sedangkan skala besar, SSR dapat menghasilkan citra lebih alami. Tetapi SSR tidak dapat melakukan kedua hal tersebut sekaligus.
2. Jika SSR digunakan dalam skala rendah maka akan lebih sering memunculkan area berwarna serupa menjadi berwarna abu-abu.

Ide dari dasar *Multiscale Retinex* pada masalah SSR terdapat pada skala berbeda yang digunakan dan diberi bobot yang tidak serupa untuk menggabungkan kelebihan dan meniadakan kelemahan yang dimiliki dari skala rendah dan skala besar. Berikut adalah formulasi original *MultiScale Retinex*.

$$R_{msri} = \sum_{n=1}^N W_n R_{ni}$$

R_{msri} Merupakan hasil dari *Multiscale Retinex* (MSR) ialah hasil dari jumlah SSR yang masing-masing telah diberi bobot. N ialah jumlah skala yang dipakai. W_n adalah bobot yang diasosiasikan dengan skala ke- n . R_{ni} merupakan output dari SSR yang diasosiasikan dengan skala ke- n . Simbol i menyatakan channel warna, misalnya Merah(R), Hijau(G), dan Biru(B). Ada beberapa parameter dalam MSR antara lain :

1. Jumlah Skala (N)

Berdasarkan waktu komputasi yang cepat jumlah skala yang dipakai adalah 3 jumlah, sebagai jumlah skala minimum yang menyediakan output yang baik.

2. Sigma yang digunakan (σ_n)

Sigma dalam parameter terdiri dari tiga kombinasi yaitu tinggi, sedang, dan rendah.

3. Bobot untuk setiap input SSR (W_n)

Pembobotan dalam proses MSR hasilnya harus = 1. Contohnya jika $N=3$, maka $W_n = W_1+W_2+W_3$ dimana $W_n = 1$

2.5 Median Filter

Median filter adalah filter non linear yang dikembangkan oleh Tukey. Fungsi metode ini untuk pengurangan noise dan menghaluskan suatu citra. Sistem kerja penapis ini tidak termasuk kedalam bagian operasi *konvolusi*, Oleh sebab itu disebut filter non linear. operasi non linear dihitung kemudian mengurutkan nilai intensitas *pixel* yang diproses dengan nilai tertentu (Maulana et al., 2016).

Suatu citra yang memiliki sejumlah *pixel* ganjil digeser titik per titik pada daerah seluruh citra. Kemudiannilai tersebut diurutkan dari yang terendah hingga tertinggi kemudian dihitung nilai tengahnya. Nilai tengah tersebut dapat menggantikan nilai yang ada pada bidang pusat window.

median filter adalah metode yang berfokus terhadap nilai tengah, dari jumlah total nilai keseluruhan pixel yang ada pada citra, contoh-Nya data $a=4$, $b=7$, $c=9$, $d=11$, dan $e=13$, maka proses dari *median filter* ialah mencari nilai tengah dari keseluruhan data yang sudah diurutkan dari nilai yang paling rendah hingga pada data nilai yang tertinggi kemudian diambil nilai tengahnya (4, 7, 9, 11, 13). Maka nilai tengah pada data tersebut 9 (Nur et al., 2018).

Proses *median filter* teknik yang dilakukan yaitu dengan mencari nilai tengah dari nilai pixel tetangganya yang mempengaruhi nilai tengah. Cara kerja teknik ini yaitu mengisi nilai *pixel* dari setiap *pixel* dengan nilai media tetangganya. Mengurutkan nilai-nilai *pixel* tetangga, lalu dipilih nilai tengahnya merupakan Proses awal dari median filter.

Filter median sangat terkenal hal ini dikarenakan untuk jenis random *noise* tertentu memberikan kemampuan penurunan *noise* yang sangat baik, dengan memperhatikan pengurangan *bluring filter something linier* pada ukuran yang sama. Filter median tertentu menunjukkan adanya *noise*.



Gambar 2.4 Citra sampel

Gambar diatas merupakan citra yang berukuran 5 x 5 pixel dimana citra 5 x 5 ini akan menjadi sample dalam perhitungan metode *median filter*.

Tabel 2.1 Citra Input

73	96	107	122	120
116	109	114	126	114
116	111	124	126	127
130	116	121	135	124
106	99	109	122	131

Perhitungan menggunakan metode *median filter*, suatu metode yang dapat menghilangkan *noise*. Cara kerja *median filter* yaitu dengan mencari nilai tengah dari nilai *pixel*, perhitungannya dengan cara mencari nilai dari kolom pertama hingga kolom ketiga berdasarkan pergeseran kernel.

Tabel 2.2 Proses pencarian kernel/mask

73	96	107	122	120
116	109	114	126	114
116	111	124	126	127
130	116	121	135	124
106	99	109	122	131

Kernel yang memiliki ordo baris dan kolom sama. Seperti contoh sample tersebut 3 x 3, dengan dimensi yang kecil yang berfungsi untuk pemrosesan konvolusi. Perhitungannya dengan cara menyusun dari angka terendah hingga

angka tertinggi, agar pengambilan nilai *pixel* dari nilai angka *pixel* yang paling tengah.

Kernel/mask 3x3			
73	96	107	F(x) = 73, 96, 107, 109, (111)114, 116, 116, 124
116	109	114	
116	111	124	

Hasil pencarian dari kesembilan kernel/mask tersebut dimasukkan kedalam nilai tengah *pixel*.

Tabel 2.3 Hasil Output 5x5 pixel

73	96	107	122	120
116	111	114	122	114
116	116	121	124	127
130	116	121	124	124
106	99	109	122	131



Gambar 2.5 Citra Asli

Gambar diatas merupakan citra asli dari citra sample, terlihat pada citra *noise/derau* sebelum dilakukannya proses *median filter*.



Gambar 2.6 Hasil Citra Output

Gambar diatas merupakan citra hasil dari proses *median filter*, dimana pada citra sudah terjadi pengurangan noise/derau yang sebelumnya ada pada Gambar 2.5.

2.6 *Histogram*

Histogram merupakan jenis diagram yang menampilkan distribusi frekuensi nilai intensitas *pixel* pada suatu citra. pada sumbu X merupakan tingkat kecerahan warna, sedangkan pada sumbu Y merupakan frekuensi atau jumlah kemunculan *pixel* suatu citra, sumbu horizontal diagram merupakan nilai grey-level dari citra dan sumbu vertical merupakan jumlah kemunculan *pixel*. Histogram mempunyai manfaat, (Sutoyo, 2009) yaitu:

1. Untuk mempermudah dalam menentukan tingkat keabuan yang sesuai agar dapat memperoleh citra yang berkualitas baik sesuai yang diharapkan.
Contoh : kecermerlangan, pengubahan kontras, dan lain sebagainya.
2. Untuk pemilihan batas ambang (threshold)
Misalnya : pada segmentasi citra dimana adanya pemisahan objek dari latar belakangnya, pada proses ini agar objek dan latar belakang terpisahkan maka harus menentukan batas-batas nilai keabuan dari objek dan batas-batas nilai keabuan latar belakangnya.

2.7 *Matlab*

Matlab (*Matrix Laboratory*) yaitu suatu aplikasi yang digunakan sebagai analisis dan komputasi numeric yang merupakan suatu bahasa pemrograman matematika lanjutan yang dibangun dengan dasar pemikiran menggunakan sifat dan bentuk matriks. Pada awalnya, program ini adalah suatu *interface* untuk koleksi rutin-rutin numeric pada proyek *LINPACK* dan *EISPACK*, yang dikembangkan menggunakan bahasa *FORTRAN* namun pada saat initermasuk dalam produk komersial dari perusahaan *Mathworks.Inc*, lalu pada perkembangan selanjutnya dikembangkan menggunakan bahasa C++ dan assembler (utamanya untuk fungsi-fungsi dasar *Matlab*), (Cahyono, n.d.).

Matlab adalah label software yang dikembangkan oleh *Mathworks.Inc* termasuk dalam software yang paling efisien untuk memperhitungkan numeric berbasis matriks. Maka dari itu ketika saat perhitungan kita bisa memformulasikan suatu permasalahan kedalam bentuk matriks, maka aplikasi *matlab* adalah software yang sangat cocok untuk penyelesaian numericnya. *Matlab* adalah software yang menggunakan bahasa tingkat tinggi berbasis pada matriks yang sering dimanfaatkan dalam teknik komputasi numeric, agar dapat membantu suatu permasalahan yang melibatkan optimasi, aproksimasi, operasi matematika elemen, matriks, dan lain sebagainya. Ada beberapa pembelajaran yang menggunakan Matlab yaitu :





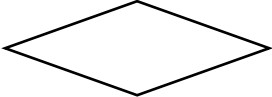

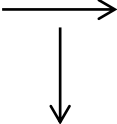

- 1) Pengembangan dan Algoritma
- 2) Matematika dan komputansi
- 3) Pengembangan aplikasi teknik
- 4) Eksplorasi, visualisasi dan Analisa data,
- 5) Analisis numeric dan statistic
- 6) pembuatan prototype, simulasi dan Pemrograman Modeling.



2.8 Flowchart

Flowchart merupakan suatu simbol yang menunjukkan tahapan dari suatu program secara mendetail. *Flowchart* berfungsi untuk menjabarkan jalan dari program yang sudah dikerjakan dalam sistem grafis sehingga pengguna lain dapat mengerti dan memahami program yang sudah dirancang. Didalam *flowchart* terdiri dari beberapa jenis simbol seperti penerimaan masukan, pemrosesan, dan terakhir adalah hasil, didalam suatu masalah harus ada simbol masukan, dan pemrosesan karena ini saling menghubungkan agar bisa mendapatkan hasilnya (Suprpto, 2008)

Beberapa jenis simbol yang selalu dipakai dalam penggunaan *flowchart* pada suatu program yang telah dirancang antara lain.

Tabel 2.4 Simbol-simbol Flowchart

Simbol	Keterangan
	Masukan, Merupakan suatu tahapan dalam menerima data, dimana pada simbol ini awal mula data suatu masalah dimasukkan untuk selanjutnya diproses.
	Masukan manual, di input menggunakan <i>keyboard</i> , ataupun perangkat masukan lain seperti halnya <i>barcode reader</i> .
	Proses, data yang dimasukkan selanjutnya dilakukan proses untuk memperoleh jawaban atas persoalan yang ingin diselesaikan.
	Keluaran, merupakan hasil dari suatu data yang telah di proses.
	Simbol ini adalah suatu kegiatan yang dilakukan dalam pengecekan suatu kondisi apakah sudah sesuai atau tidak sesuai.
	Sub rutin terdiri dari dua jenis yaitu <i>procedure</i> dan <i>function</i> . Perbedaanannya, ketika dipanggil prosedur tidak mengembalikan suatu nilai tetapi fungsi mengembalikannya.
	Simbol ini adalah arah sebagai petunjuk untuk menghubungkan tahapan ataupun proses yang selanjutnya akan dilakukan dalam suatu masalah pada <i>flowchart</i>
	Terminator, berfungsi untuk memulai dan mengakhiri suatu masalah yang ada pada <i>flowchart</i> , seperti label mulai dan selesai

	Simbol ini berguna sebagai penghubung antara satu tahapan dengan tahapan lain dalam sebuah <i>flowchart</i> .
	Simbol ini merupakan hasil dari suatu pemecahan masalah yang sudah dicetak (print out).

2.9 Penelitian Terkait

Pengembangan pengolahan citra dalam melakukan perbaikan kualitas citra atau *Image Enhancement* telah banyak yang melakukan penelitian membahas tentang ini diantaranya oleh Deni Supriadi dan Panji Novantara (2017). Melakukan penelitian implementasi metode *Multiscale Retinex* untuk *image enhancement* menggunakan *delphi* yang bertujuan untuk dalam melakukan perbaikan pencerahan dari sebuah citra sehingga pada bagian yang terlalu gelap menjadi terang dengan menggunakan metode *Multiscale Retinex*.

Selain itu, (Harahap, 2018) juga melakukan penelitian yang berjudul Implementasi metode *retinex* untuk meningkatkan kualitas citra underwater yang bertujuan untuk meningkatkan kualitas citra bawah laut yang cenderung kebiruan dan gelap.

Dan juga penelitian (Dwi Prasetyo, 2018) yang berjudul Peningkatan ekstraksi fitur berbasis scale invariant feature transform menggunakan metode *Multiscale Retinex* untuk meningkatkan jumlah keypoint, yang bertujuan meningkatkan jumlah keypoint pengenalan pola.

Berdasarkan Penelitian diatas untuk melakukan perbaikan kualitas citra atau *Image Enhancement* dengan menggunakan metode-metode *Image Enhancement*, seperti *Multiscale Retinex* dan *Median Filter* dapat membantu pengguna agar dapat memanfaatkan hasil perbaikan citra itu sendiri sesuai dengan kebutuhan.

BAB III

METODOLOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

3.1.1 Tempat Penelitian

Tempat dan waktu penelitian dilakukan di Laboratorium Multimedia Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara.

3.1.2 Waktu & Jadwal Pelaksanaan Penelitian

Waktu dan jadwal pelaksanaan penelitian ini dilaksanakan pada semester genap tahun ajaran 2020.

3.2 Bahan dan Alat Penelitian

Dalam penelitian ini penulis memerlukan alat dan bahan yang sesuai dan mendukung judul penelitian agar dapat membantu dan mempermudah pelaksanaan penelitian.

3.2.1 Perangkat Keras

Perangkat keras yang digunakan pada pengembangan sistem ini diperlukan sebagai berikut Laptop, Processor Intel(R) Core(TM) i5-2410M CPU@2.30GHz, Installed Memory (RAM) 4.00 GB, dan Harddisk 500 GB.

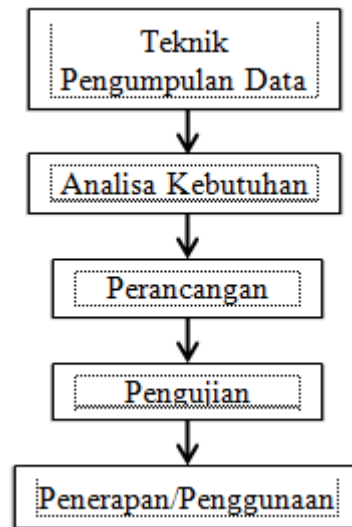
3.2.2 Perangkat Lunak

1. Operating System Windows 10 Pro 64 bit
2. Matlab 2015

3.3 Cara Kerja

3.3.1 Perencanaan

Perencanaan dalam penelitian ini dilakukan melalui tahap-tahap yang dilakukan sebagai berikut :



Gambar 3.1 Tahap-tahap Perencanaan

3.3.2 Teknik Pengumpulan Data

Teknik pengumpulan data yang dilakukan dalam penelitian ini adalah sebagai berikut :

1. Perpustakaan (*library*)

Perpustakaan termasuk ke dalam salah satu tempat pengumpulan data dimana didalam perpustakaan terdapat berbagai macam buku-buku yang bisa dijadikan referensi, sesuai data apa yang ingin diambil oleh penulis, tentunya perpustakaan ini sangat membantu dari segi pengumpulan data karena tentu saja data yang diambil *valid* (benar) karena sudah dibukukan dan diterbitkan. Penulis mencari buku-buku diperpustakaan daerah (Pusda) Medan, perpustakaan Kampus Harapan Medan dan Perpustakaan Universitas Islam Negeri Medan, Serta mencari Paper dan Ebook pada penelitian terkait sebelumnya diinternet.

2. Studi Literatural

Studi Literatural adalah teknik yang dipakai peneliti dalam mengumpulkan kasus atau permasalahan yang sedang diteliti dari berbagai sumber-sumber seperti buku, jurnal, artikel laporan penelitian yang telah dibuat sebelumnya. Sumber-sumber tersebut nantinya akan menjadi rujukan yang bisa

dipertanggung jawabkan penulis mengenai hasil dari penelitian yang dilakukan.

3. Kamera Digital

Kamera digital digunakan untuk mengumpulkan data citra digital yang ingin diteliti, baik itu citra pemandangan, wajah, maupun citra objek-objek tertentu lainnya yang memiliki masalah kualitas citra.

3.3.3 Analisa Kebutuhan

Analisa kebutuhan merupakan proses dalam memperoleh suatu gambaran ataupun rancangan perangkat lunak yang dibutuhkan oleh pengguna. Analisa kebutuhan ini adalah suatu cara yang sangat berpengaruh pada saat identifikasi masalah-masalah yang timbul pada saat melakukan penelitian. Dalam melakukan analisis kebutuhan diperlukan adanya perancangan, pengumpulan data dan analisis data.

Dalam suatu citra digital masih banyak terdapat masalah yang terjadi terhadap kualitas citra, tidak selamanya citra yang dihasilkan sesuai dengan keinginan. Derau pada citra dapat mempengaruhi kualitas citra seperti kabur, degradasi atau penurunan mutu, dan warna terlalu terang. Gangguan pada citra digital ini dikarenakan adanya berbagai faktor, misalnya kurang pencahayaan ketika dilakukan pemotretan gambar, terbatasnya spesifikasi dari kamera yang dipakai, atau terbatasnya kondisi dalam mengambil gambar.

Tujuan dilakukannya penelitian ini yaitu untuk menerapkan sistem pengolahan citra dengan menggunakan Metode *Multiscale Retinex* dan *Median Filter* agar mempermudah proses memperbaiki kualitas citra digital yang memiliki permasalahan dalam kualitas citra, Agar hasil citra sesuai dengan yang diharapkan. Karena hasil dari sistem pengolahan citra ini dapat membantu menyelesaikan permasalahan tersebut. Ada beberapa batasan dalam menjalankan sistem aplikasi ini, yaitu :

1. Sistem akan melakukan perbaikan citra digital berformat*.jpg yang memiliki kualitas rendah dengan intensitas cahaya yang kurang.

2. Sistem melakukan perbaikan citra digital dengan menggunakan metode *Multiscale Retinex* dan *Median Filter*.

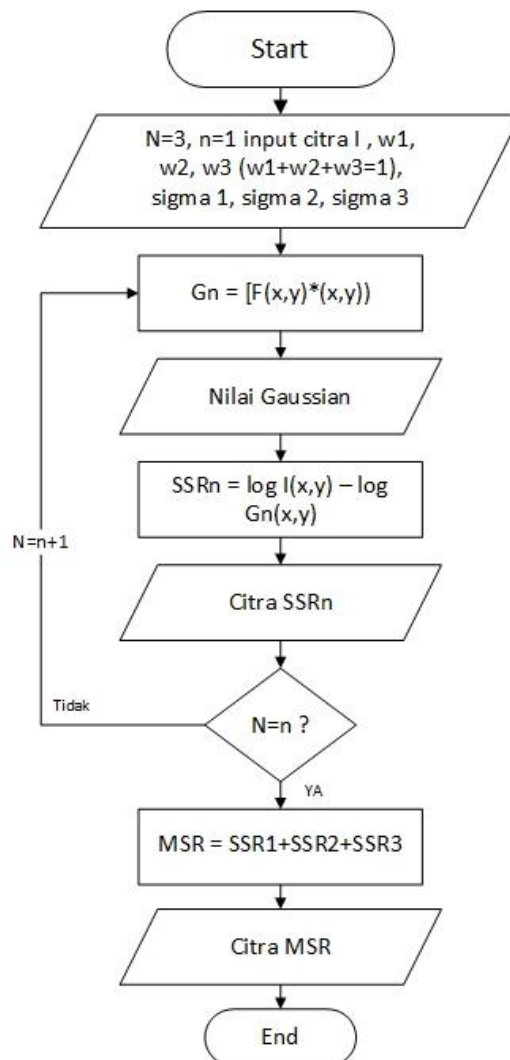
3.3.4 Perancangan

3.3.4.1 *Flowchart* merupakan suatu skema dengan menggunakan berbagai simbol yang memberikan penggambaran suatu alur dari tahapan-tahapan yang saling berhubungan pada suatu permasalahan/program. Berikut ini merupakan *flowchart* dari metode *multiscale retinex* dan *median filter* yang digunakan dalam perbaikan kualitas citra digital.



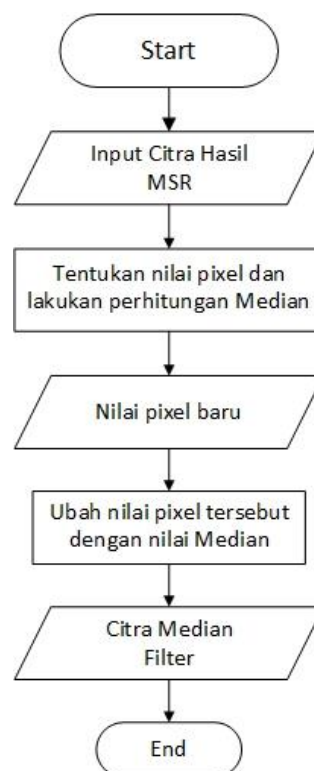
Gambar 3.2 Flowchart Sistem

Gambar diatas merupakan alur dari sebuah sistem dalam melakukan perbaikan kualitas citra digital dengan metode *Multiscale Retinex* dan *Median Filter*, dimulai dari memasukkan citra digital berformat *.jpg kemudian dilakukan perbaikan dengan metode *Single Scale Retinex* (SSR) selanjutnya citra dari hasil SSR akan dilakukan ketahap perbaikan dengan metode *Multisale Retinex* (MSR) hasil dari citra MSR akan dilakukan penghalusan citra dengan metode *Median Filter*. Maka citra hasil perbaikan kualitas citra digital dengan metode *Multisale Retinex* dan *Median Filter* keluar dan selesai.



Gambar 3.3 Flowchart Multiscale Retinex

Gambar diatas merupakan algoritma dari metode *Multiscale Retinex*, dimulai dari input dengan keterangan $N=3$ merupakan jumlah skala yang dipakai, $n=1$ untuk melakukan langkah SSR pertama, masukkan citra agar diproses, W adalah nilai weight ketika $W1+W2+W3$ dijumlahkan maka hasilnya sama dengan 1, sigma 1, sigma 2 dan sigma 3 merupakan sigma yang dipakai terdiri dari nilai rendah, sedang dan tinggi, selanjutnya mencari nilai gaussian rumus yang dipakai yaitu $Gn = [F(x,y)*[x,y]]$, lalu melakukan langkah SSR, rumusnya yaitu $SSRn = \log In(x,y) - \log Gn(x,y)$. maka akan mendapatkan citra hasil $SSRn$, Jika $N = (n=1)$ maka akan mengulangi proses mencari Gaussian dengan n ditambah 1, sampai mengulangi tiga kali proses sehingga didapat tiga *Singlescale Retinex* maka akan masuk ke langkah *Multiscale Retinex*, Selanjutnya langkah *Multiscale Retinex* dengan menjumlahkan hasil $SSR1, SSR2$ dan $SSR3$, maka akan mendapatkan hasil citra *Multiscale Retinex*, Selesai.



Gambar 3.4 Flowchart Median Filter

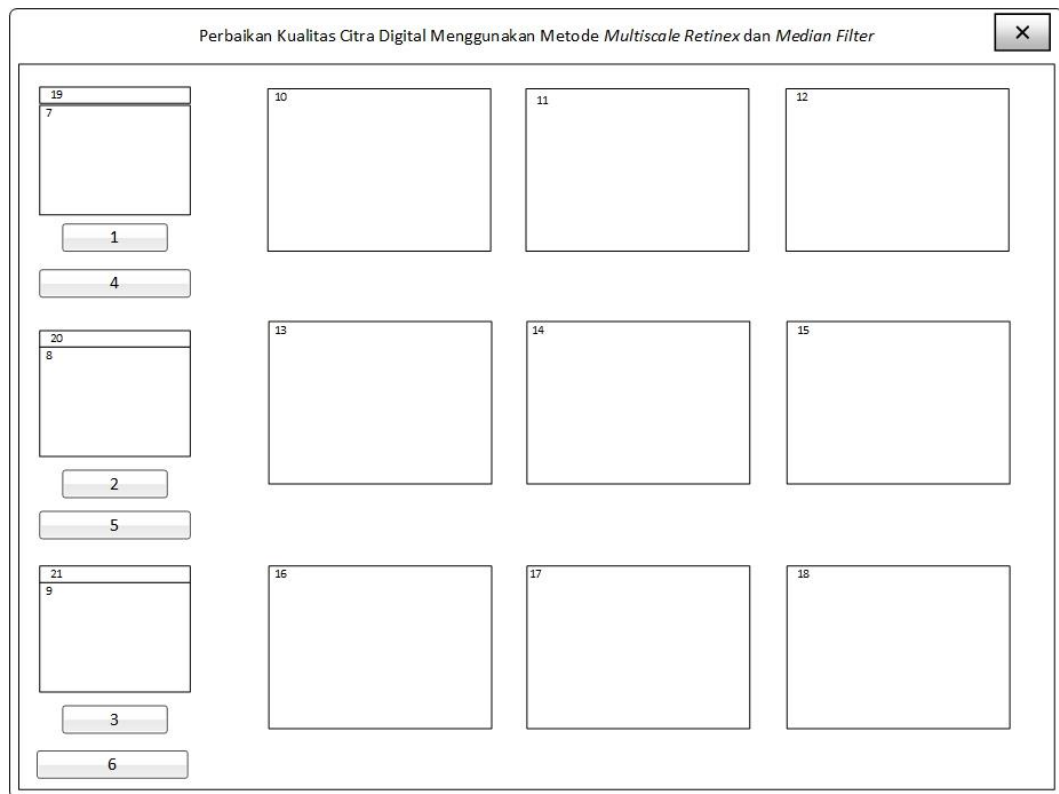
Gambar diatas merupakan proses dari median filter, dimulai dari input citra hasil dari citra *Multiscale Retinex*, kemudian diproses dengan formula $f(x,y)=\text{median}\{g(s,t)\}$ lalu dilakukan perhitungan dengan mencari nilai tengah dari nilai pixel pada citra, dengan mengurutkan nilai pixel dari nilai terendah sampai nilai pixel tertinggi, sehingga mendapatkan nilai median, kemudian akan didapat nilai pixel baru dari hasil perhitungan median tersebut sehingga di dapat citra hasil *median filter*, selesai.

3.3.4.2 Perancangan Dialog

Gambar 3.5 Rancangan Dialog

Gambar diatas merupakan rancangan dialog dalam melakukan perbaikan kualitas citra digital dengan menggunakan Metode *Multiscale Retinex* dan *Median Filter* dengan keterangan :

- 1) Suatu *Axes* yang berfungsi memunculkan citra awal.
- 2) Suatu *Axes* yang berfungsi memunculkan hasil citra *Singlescale Retinex1*.
- 3) Suatu *Axes* yang berfungsi memunculkan hasil citra *Singlescale Retinex2*.
- 4) Suatu *Axe* syang berfungsi memunculkan hasil citra *Singlescale Retinex3*.
- 5) Suatu *Axes* yang berfungsi memunculkan citra hasil *Multiscale Retinex*.
- 6) Suatu *Axes* yang berfungsi memunculkan citra hasil *Median Filter*.
- 7) Merupakan *Button* untuk menginput citra asli.
- 8) Merupakan *Button* proses *Multiscale Retinex*.
- 9) Merupakan *Buttons* impan citra hasil *Multiscale Retinex*.
- 10) Merupakan *Button* Proses *Median Filter*.
- 11) Merupakan *Button* Simpan *Median Filter*.
- 12) Merupakan *Buton* untuk membersihkan program yang berjalan.
- 13) Merupakan *Label* citra asli.
- 14) Merupakan *Label* Citra *Multiscale Retinex*.
- 15) Merupakan *Label* Citra *Median Filter*.
- 16) Merupakan *Label* *Singelscale Retinex1*.
- 17) Merupakan *Label* *Singlescale Retinex2*.
- 18) Merupakan *Label* *Singlescale Retinex3*.



Gambar 3.6 Form Histogram

Gambar diatas merupakan tampilan *Histogram* dari citra asli (R,G,B), Citra *Multiscale Retinex* (R,G,B) dan citra *Median Filter* (R,G,B). Dimana *histogram* nantinya akan berbentuk diagram yang menampilkan nilai kemunculan intensitas citra dan menggambarkan penyebaran nilai-nilai intensitas *pixel* dari suatu citra, pada titik koordinat y adalah nilai intensitas *pixel* sedangkan pada titik korrdinat x adalah frekuensi/jumlah *pixel*, dengan keterangan :

- 1) Sampai 3) Merupakan *Button* untuk menginpu citra
- 4) Sampai 6) Merupakan *Button histogram*
- 7) Sampai 9) Merupakan *Axes* Untuk menampilkan Citra Asli, Msr dan Median Filter.
- 10) Sampai 18) Merupakan tampilan grafik *histogram* dengan sumbu x,y
- 19) Sampai 21) Merupakan Label pada masing – masing citra.

3.3.5 Pengujian

Pengujian sistem ini dilakukan untuk menguji kinerja sistem yang telah dibuat dan mengetahui tingkat keberhasilan dan kegagalan dari sebuah sistem yang telah dirancang sebagai sistem perbaikan kualitas citra digital. Sistem akan diujikan terhadap beberapa citra digital yang minim intensitas cahaya dan juga mengalami derau. Pengujian sistem dilakukan dengan beberapa tahapan sesuai dengan langkah-langkah proses dari metode. Setiap metode dari penelitian memiliki proses dalam mengelemisasi citra digital. Pengujian akan dimulai dari penjelasan mengenai rancangan suatu sistem yang telah dibuat seperti menu-menu yang ada pada rancangan sistem.

Pengujian dilakukan dengan penginputan citra digital, citra yang diinput harus memenuhi syarat agar bisa diproses oleh sistem, kemudian sistempengolahan citra akan memproses berdasarkan metode yang telah digunakan yaitu metode *Multisale Retinex* yang berfungsi untuk mencerahkan citra, selanjutnya akan dilanjutkan dengan metode *Median Filter* untuk menghilangkan derau/ noise, setelah itu akan terlihat hasil dari citra, dan akan terlihat perbandingan citra asli dan citra hasil dari sistem yang telah dibuat.

3.3.6 Penerapan/Penggunaan

Penerapan/penggunaan sistem untuk memperbaiki kualitas citra digital, pada citra digital yang mempunyai kualitas rendah akan di proses menjadi citra digital yang mempunyai kualitas lebih baik, agar hasil citra terlihat lebih jelas dan terang. Dengan menggunakan pengolahan citra dan menerapkan metode *Multiscale Retinex* dan *Median Filter*, yang bertujuan untukmemberi peningkatan kualitas citra digital sesuai dengan yang kita harapkan.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pembahasan

Didalam Pembahasan yang akan dilakukan dalam penelitian ini terdapat proses-proses yang harus dikerjakan, sehingga hasil dalam penelitian yang diinginkan tercapai, antara lain :

4.1.1 Analisis Data

Data yang dibutuhkan pada penelitian ini adalah data berupa citra digital yang minim intensitas cahaya, yang kemudian diolah dengan menggunakan pengolahan citra menggunakan metode *Multiscale Retinex* dan *Median Filter*. didalam proses analisis data dilakukan pemilihan citra yang sesuai dengan batasan masalah. Kemudian menerapkan sistem aplikasi dengan metode *Multiscale Retinex* dan *Median Filter* untuk melakukan perbaikan kualitas citra digital menggunakan matlab 2015. Langkah-langkah dalam menjalankan sistem aplikasi antara lain :

- a. Menginput citra digital berformat jpg.
- b. Melakukan proses *Singlescale Retinex*.
- c. Selanjutnya perbaikan kecerahan citra dengan metode *Multiscale Retinex*.
- d. Pengurangan derau/noise dengan metode *Median Filter*.

4.1.2 Representasi Data

Proses pertamayangakan dilakukan yaitu proses *Singlescale Retinex* (SSR), setelah hasil didapat selanjutnya ke proses *Multiscale Retinex* (MSR), untuk proses perbaikan citra.

- a. Metode *Singlescale Retinex*

Proses *Singlescale Retinex* (SSR) merupakan proses awal sebelum proses *Multiscale Retinex* (MSR), dengan melalui tiga tahap SSR. Banyak jurnal yang menyebutkan bahwa *Singlescale Retinex* (SSR) merupakan *Retinex* skala tunggal

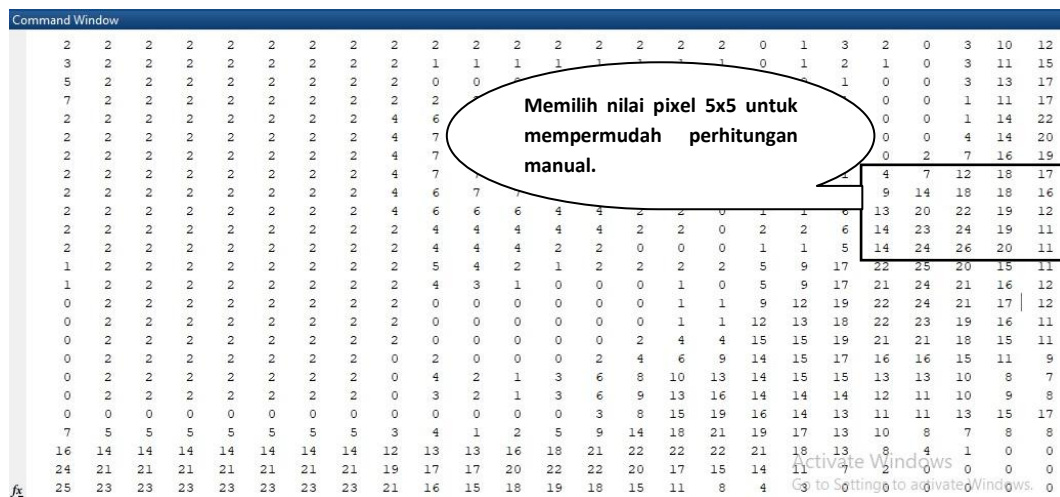
karena hanya menggunakan satu fungsi surround untuk mencari tahu iluminasi dengan gambar yang berbelit-belit, berikut rumus SSR :

$$SSRn(x,y) = \log In(x,y) - \log Gn(x,y)$$

Selanjutnya melakukan konversi citra digital kedalam matriks untuk mendapatkan nilai *pixel* dengan menggunakan bantuan matlab, dimana ukuran matriks yang digunakan adalah 5x5. Adapun matriks citra yang dibentuk adalah ;



Gambar 4.1 Citra Asli



Gambar 4.2 Potongan Nilai Pixel Citra RGB

Adapun contoh perhitungan manual yaitu :

Tahap awal yang dikerjakan ialah melakukan perhitungan untuk mendapatkan nilai *Gaussian*, yaitu :

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Keterangan : σ = sebuah sigma

$$\pi = \frac{22}{7}$$

$$e = 2,71828182846$$

- Misalnya $\sigma = 8,5$

$$\begin{pmatrix} \frac{1}{2\pi(8,5)^2} e^{-\frac{-1^2+1^2}{2(8,5)^2}} & \frac{1}{2\pi(8,5)^2} e^{-\frac{0^2+1^2}{2(8,5)^2}} & \frac{1}{2\pi(8,5)^2} e^{-\frac{1^2+1^2}{2(8,5)^2}} \\ \frac{1}{2\pi(8,5)^2} e^{-\frac{-1^2+0^2}{2(8,5)^2}} & \frac{1}{2\pi(8,5)^2} e^{-\frac{0^2+0^2}{2(8,5)^2}} & \frac{1}{2\pi(8,5)^2} e^{-\frac{1^2+0^2}{2(8,5)^2}} \\ \frac{1}{2\pi(8,5)^2} e^{-\frac{-1^2+-1^2}{2(8,5)^2}} & \frac{1}{2\pi(8,5)^2} e^{-\frac{0^2+-1^2}{2(8,5)^2}} & \frac{1}{2\pi(8,5)^2} e^{-\frac{1^2+-1^2}{2(8,5)^2}} \end{pmatrix}$$

$$\begin{pmatrix} 0.0021765393 & 0.0021887541 & 0.0021765393 \\ 0.0021887541 & 0.0022039538 & 0.0021887541 \\ 0.0021765393 & 0.0021887541 & 0.0021765393 \end{pmatrix}$$

Total elemen matriks: 0.0196651274

Total Keseluruhan elemen matriks setelah dijumlahkan hasilnya berjumlah 1, jika tidak bernilai 1 maka dilakukan perhitungan ulang, rumus yang dipakai sebagai berikut :

$$New\alpha = \alpha x \frac{1}{Total}$$

$$New\alpha_{(1,1)} = 0.0021765393 x \frac{1}{0.0196651274} = 0,110670999$$

$$New\alpha_{(1,2)} = 0.0021887541 x \frac{1}{0.0196651274} = 0,111301293$$

$$New\alpha_{(1,3)} = 0.0021765393 x \frac{1}{0.0196651274} = 0,110670999$$

$$New\alpha_{(2,1)} = 0.0021887541 x \frac{1}{0.0196651274} = 0,111301293$$

$$New\alpha_{(2,2)} = 0.0022039538 x \frac{1}{0.0196651274} = 0,112074219$$

$$New\alpha_{(2,3)} = 0.0021887541 x \frac{1}{0.0196651274} = 0,111301293$$

$$New\alpha_{(3,1)} = 0.0021765393 x \frac{1}{0.0196651274} = 0,110670999$$

$$New\alpha_{(3,2)} = 0.0021887541 x \frac{1}{0.0196651274} = 0,111301293$$

$$New\alpha_{(3,3)} = 0.0021765393 x \frac{1}{0.0196651274} = 0,110670999$$

lakukan perhitungan pada seluruh elemen matriks, agar mendapatkan hasil seperti dibawah ini :

$$\begin{pmatrix} 0,110670999 & 0,111301293 & 0,110670999 \\ 0,111301293 & 0,112074219 & 0,111301293 \\ 0,110670999 & 0,111301293 & 0,110670999 \end{pmatrix}$$

Nilai konvolusi yang dihasilkan merupakan perhitungan nilai *pixel* citra awal dan kernel *Gaussian*. langkah selanjutnya ialah operasi konvolusi dengan proses memindahkan kernel pixel perpixel lalu dilakukan perkalian dan penjumlahan dengan nilai Gaussian agar mendapatkan hasil Konvolusi. Yaitu $f(x,y)*G(x,y)$.

0	0	0	0	0	0	0
0	4	7	12	18	17	0
0	9	14	18	18	16	0
0	13	20	22	19	12	0
0	14	23	24	19	11	0
0	14	24	26	20	11	0
0	0	0	0	0	0	0

F(x,y)

0,11067099	0,11130129	0,11067099
0,11130129	0,11207421	0,11130129
0,11067099	0,11130129	0,11067099

G(x,y)

Untuk mendapatkan nilai konvolusi maka dilakukan perhitungan dengan nilai matriks F(x,y) dan nilai Gaussian G(x,y), dimulai dari sudut kiri atas matriks.

0	0	0	0	0	0	0
0	4	7	12	18	17	0
0	9	14	18	18	16	0
0	13	20	22	19	12	0
0	14	23	24	19	11	0
0	14	24	26	20	11	0
0	0	0	0	0	0	0

→

3,77851134				

- Perhitungan yang dilakukan :

$$(0 \times 0,11067099) + (0 \times 0,11130129) + (0 \times 0,11067099) + (0 \times 0,11130129) + (4 \times 0,11207421) + (7 \times 0,11130129) + (0 \times 0,11067099) + (9 \times 0,11130129) + (14 \times 0,11067099) = 3,77851134$$

Selanjutnya geser kernel satu *pixel* kekanan, kemudian lakukan perhitungan pada posisi (0,0) dari kernel, seperti yang sebelumnya.

0	0	0	0	0	0	0
0	4	7	12	18	17	0
0	9	14	18	18	16	0
0	13	20	22	19	12	0
0	14	23	24	19	11	0
0	14	24	26	20	11	0
0	0	0	0	0	0	0

	7,1116749			

- Perhitungan yang dilakukan :

$$(0 \times 0,11067099) + (0 \times 0,11130129) + (0 \times 0,11067099) + (4 \times 0,11130129) + (7 \times 0,11207421) + (12 \times 0,11130129) + (9 \times 0,11067099) + (14 \times 0,11130129) + (18 \times 0,11067099) = 7,1116749$$

Selanjutnya geser kernel satu *pixel* kekanan, kemudian lakukan perhitungan pada posisi (0,0) dari kernel, seperti yang sebelumnya.

0	0	0	0	0	0	0
0	4	7	12	18	17	0
0	9	14	18	18	16	0
0	13	20	22	19	12	0
0	14	23	24	19	11	0
0	14	24	26	20	11	0
0	0	0	0	0	0	0

		9,67231767		

- Perhitungan yang dilakukan :

$$(0 \times 0,11067099) + (0 \times 0,11130129) + (0 \times 0,11067099) + (7 \times 0,11130129) + (12 \times 0,11207421) + (18 \times 0,11130129) + (14 \times 0,11067099) + (18 \times 0,11130129) + (18 \times 0,11067099) = 9,67231767$$

Selanjutnya geser kernel satu *pixel* kekanan, kemudian lakukan perhitungan pada posisi (0,0) dari kernel, seperti yang sebelumnya.

0	0	0	0	0	0	0
0	4	7	12	18	17	0
0	9	14	18	18	16	0
0	13	20	22	19	12	0
0	14	23	24	19	11	0
0	14	24	26	20	11	0
0	0	0	0	0	0	0

			11,01131007	

- Perhitungan yang dilakukan :

$$(0 \times 0,11067099) + (0 \times 0,11130129) + (0 \times 0,11067099) + (12 \times 0,11130129) + (18 \times 0,11207421) + (17 \times 0,11130129) + (18 \times 0,11067099) + (18 \times 0,11130129) + (16 \times 0,11067099) = 11,01131007$$

Selanjutnya geser kernel satu *pixel* kekanan, kemudian lakukan perhitungan pada posisi (0,0) dari kernel, seperti yang sebelumnya.

0	0	0	0	0	0	0
0	4	7	12	18	17	0
0	9	14	18	18	16	0
0	13	20	22	19	12	0
0	14	23	24	19	11	0
0	14	24	26	20	11	0
0	0	0	0	0	0	0

				7,68158325

- Perhitungan yang dilakukan :

$$(0 \times 0,11067099) + (0 \times 0,11130129) + (0 \times 0,11067099) + (18 \times 0,11130129) + (17 \times 0,11207421) + (0 \times 0,11130129) + (18 \times 0,11067099) + (16 \times 0,11130129) + (0 \times 0,11067099) = 7,68158325$$

berikutnya pindahkan kernel kesudut kiri bawah, kemudian lakukan perhitungan seperti sebelumnya.

0	0	0	0	0	0	0
0	4	7	12	18	17	0
0	9	14	18	18	16	0
0	13	20	22	19	12	0
0	14	23	24	19	11	0
0	14	24	26	20	11	0
0	0	0	0	0	0	0

9,7845345				

- Perhitungan yang dilakukan :

$$(0 \times 0,11067099) + (4 \times 0,11130129) + (7 \times 0,11067099) + (0 \times 0,11130129) + (9 \times 0,11207421) + (14 \times 0,11130129) + (0 \times 0,11067099) + (13 \times 0,11130129) + (20 \times 0,11067099) = 7,44712461$$

Selanjutnya geser kernel satu *pixel* kekanan, kemudian lakukan perhitungan pada posisi (0,0) dari kernel, seperti yang sebelumnya.

0	0	0	0	0	0	0
0	4	7	12	18	17	0
0	9	14	18	18	16	0
0	13	20	22	19	12	0
0	14	23	24	19	11	0
0	14	24	26	20	11	0
0	0	0	0	0	0	0

	13,22352909			

- Perhitungan yang dilakukan :

$$(4 \times 0,11067099) + (7 \times 0,11130129) + (12 \times 0,11067099) + (9 \times 0,11130129) + (14 \times 0,11207421) + (18 \times 0,11130129) + (13 \times 0,11067099) + (20 \times 0,11130129) + (22 \times 0,11067099) = 13,22352909$$

Selanjutnya geser kernel satu *pixel* kekanan, kemudian lakukan perhitungan pada posisi (0,0) dari kernel.seperti yang sebelumnya.

0	0	0	0	0	0	0
0	4	7	12	18	17	0
0	9	14	18	18	16	0
0	13	20	22	19	12	0
0	14	23	24	19	11	0
0	14	24	26	20	11	0
0	0	0	0	0	0	0

		16,44616428		

- Perhitungan yang dilakukan :

$$(7 \times 0,11067099) + (12 \times 0,11130129) + (18 \times 0,11067099) + (14 \times 0,11130129) + (18 \times 0,11207421) + (18 \times 0,11130129) + (20 \times 0,11067099) + (22 \times 0,11130129) + (19 \times 0,11067099) = 16,44616428$$

Selanjutnya geser kernel satu *pixel* kekanan, kemudian lakukan perhitungan seperti yang sebelumnya.

0	0	0	0	0	0	0
0	4	7	12	18	17	0
0	9	14	18	18	16	0
0	13	20	22	19	12	0
0	14	23	24	19	11	0
0	14	24	26	20	11	0
0	0	0	0	0	0	0

			16,89199974	

- Perhitungan yang dilakukan :

$$(12 \times 0,11067099) + (18 \times 0,11130129) + (17 \times 0,11067099) + (18 \times 0,11130129) \\ + (18 \times 0,11207421) + (16 \times 0,11130129) + (22 \times 0,11067099) + (19 \times 0,11130129) + (12 \times 0,11067099) = 16,89199974$$

Selanjutnya geser kernel satu *pixel* kekanan, kemudian lakukan perhitungan pada posisi (0,0) dari kernel, seperti yang sebelumnya.

0	0	0	0	0	0	0
0	4	7	12	18	17	0
0	9	14	18	18	16	0
0	13	20	22	19	12	0
0	14	23	24	19	11	0
0	14	24	26	20	11	0
0	0	0	0	0	0	0

				11,11917462

- Perhitungan yang dilakukan :

$$(18 \times 0,11067099) + (17 \times 0,11130129) + (0 \times 0,11067099) + (18 \times 0,11130129) \\ + (16 \times 0,11207421) + (0 \times 0,11130129) + (19 \times 0,11067099) + (12 \times 0,11130129) \\ + (0 \times 0,11067099) = 11,11917462$$

berikutnya pindahkan kernel kesudut kiri bawah, kemudian lakukan perhitungan seperti sebelumnya.

0	0	0	0	0	0	0
0	4	7	12	18	17	0
0	9	14	18	18	16	0
0	13	20	22	19	12	0
0	14	23	24	19	11	0
0	14	24	26	20	11	0
0	0	0	0	0	0	0

10,33774683				

- Perhitungan yang dilakukan :

$$(0 \times 0,11067099) + (9 \times 0,11130129) + (14 \times 0,11067099) + (0 \times 0,11130129) + (13 \times 0,11207421) + (20 \times 0,11130129) + (0 \times 0,11067099) + (14 \times 0,11130129) + (23 \times 0,11067099) = 10,33774683$$

Selanjutnya geser kernel satu *pixel* kekanan, kemudian lakukan perhitungan pada posisi (0,0) dari kernel, seperti yang sebelumnya.

0	0	0	0	0	0	0
0	4	7	12	18	17	0
0	9	14	18	18	16	0
0	13	20	22	19	12	0
0	14	23	24	19	11	0
0	14	24	26	20	11	0
0	0	0	0	0	0	0

	17,44879143			

- Perhitungan yang dilakukan :

$$(9 \times 0,11067099) + (14 \times 0,11130129) + (18 \times 0,11067099) + (13 \times 0,11130129) + (20 \times 0,11207421) + (22 \times 0,11130129) + (14 \times 0,11067099) + (23 \times 0,11130129) + (24 \times 0,11067099) = 17,44879143$$

Selanjutnya geser kernel satu *pixel* kekanan, kemudian lakukan perhitungan pada posisi (0,0) dari kernel, seperti yang sebelumnya.

0	0	0	0	0	0	0
0	4	7	12	18	17	0
0	9	14	18	18	16	0
0	13	20	22	19	12	0
0	14	23	24	19	11	0
0	14	24	26	20	11	0
0	0	0	0	0	0	0

		19,67069037		

- Perhitungan yang dilakukan :

$$(14 \times 0,11067099) + (18 \times 0,11130129) + (18 \times 0,11067099) + (20 \times 0,11130129) + (22 \times 0,11207421) + (19 \times 0,11130129) + (23 \times 0,11067099) + (24 \times 0,11130129) + (19 \times 0,11067099) = 19,67069037$$

Selanjutnya geser kernel satu *pixel* kekanan, kemudian lakukan perhitungan pada posisi (0,0) dari kernel, seperti yang sebelumnya.

0	0	0	0	0	0	0
0	4	7	12	18	17	0
0	9	14	18	18	16	0
0	13	20	22	19	12	0
0	14	23	24	19	11	0
0	14	24	26	20	11	0
0	0	0	0	0	0	0

			17,66809989	

- Perhitungan yang dilakukan :

$$(18 \times 0,11067099) + (18 \times 0,11130129) + (16 \times 0,11067099) + (22 \times 0,11130129) + (19 \times 0,11207421) + (12 \times 0,11130129) + (24 \times 0,11067099) + (19 \times 0,11130129) + (11 \times 0,11067099) = 17,66809989$$

Selanjutnya geser kernel satu *pixel* kekanan, kemudian lakukan perhitungan pada posisi (0,0) dari kernel, seperti yang sebelumnya.

0	0	0	0	0	0	0
0	4	7	12	18	17	0
0	9	14	18	18	16	0
0	13	20	22	19	12	0
0	14	23	24	19	11	0
0	14	24	26	20	11	0
0	0	0	0	0	0	0

				10,55957649

- Perhitungan yang dilakukan :

$$(18 \times 0,11067099) + (16 \times 0,11130129) + (0 \times 0,11067099) + (19 \times 0,11130129) + (12 \times 0,11207421) + (0 \times 0,11130129) + (19 \times 0,11067099) + (11 \times 0,11130129) + (0 \times 0,11067099) = 10,55957649$$

berikutnya pindahkan kernel kesudut kiri bawah, kemudian lakukan perhitungan seperti sebelumnya.

0	0	0	0	0	0	0
0	4	7	12	18	17	0
0	9	14	18	18	16	0
0	13	20	22	19	12	0
0	14	23	24	19	11	0
0	14	24	26	20	11	0
0	0	0	0	0	0	0

12,003627				

- Perhitungan yang dilakukan :

$$(0 \times 0,11067099) + (13 \times 0,11130129) + (20 \times 0,11067099) + (0 \times 0,11130129) + (14 \times 0,11207421) + (23 \times 0,11130129) + (0 \times 0,11067099) + (14 \times 0,11130129) + (24 \times 0,11067099) = 12,003627$$

Selanjutnya geser kernel satu *pixel* kekanan, kemudian lakukan perhitungan pada posisi (0,0) dari kernel, seperti yang sebelumnya.

0	0	0	0	0	0	0
0	4	7	12	18	17	0
0	9	14	18	18	16	0
0	13	20	22	19	12	0
0	14	23	24	19	11	0
0	14	24	26	20	11	0
0	0	0	0	0	0	0

	20,00473686			

- Perhitungan yang dilakukan :

$$(13 \times 0,11067099) + (20 \times 0,11130129) + (22 \times 0,11067099) + (14 \times 0,11130129) + (23 \times 0,11207421) + (24 \times 0,11130129) + (14 \times 0,11067099) + (24 \times 0,11130129) + (26 \times 0,11067099) = 20,00473686$$

Selanjutnya geser kernel satu *pixel* kekanan, kemudian lakukan perhitungan pada posisi (0,0) dari kernel, seperti yang sebelumnya.

0	0	0	0	0	0	0
0	4	7	12	18	17	0
0	9	14	18	18	16	0
0	13	20	22	19	12	0
0	14	23	24	19	11	0
0	14	24	26	20	11	0
0	0	0	0	0	0	0

		21,89258931		

- Perhitungan yang dilakukan :

$$(20 \times 0,11067099) + (22 \times 0,11130129) + (19 \times 0,11067099) + (23 \times 0,11130129) + (24 \times 0,11207421) + (19 \times 0,11130129) + (24 \times 0,11067099) + (26 \times 0,11130129) + (20 \times 0,11067099) = 21,89258931$$

Selanjutnya geser kernel satu *pixel* kekanan, kemudian lakukan perhitungan pada posisi (0,0) dari kernel, seperti yang sebelumnya.

0	0	0	0	0	0	0
0	4	7	12	18	17	0
0	9	14	18	18	16	0
0	13	20	22	19	12	0
0	14	23	24	19	11	0
0	14	24	26	20	11	0
0	0	0	0	0	0	0

			18,22334574	

- Perhitungan yang dilakukan :

$$(22 \times 0,11067099) + (19 \times 0,11130129) + (12 \times 0,11067099) + (24 \times 0,11130129) \\ + (19 \times 0,11207421) + (11 \times 0,11130129) + (26 \times 0,11067099) + (20 \times 0,11130129) + (11 \times 0,11067099) = 18,22334574$$

Selanjutnya geser kernel satu *pixel* kekanan, kemudian lakukan perhitungan pada posisi (0,0) dari kernel, seperti yang sebelumnya.

0	0	0	0	0	0	0
0	4	7	12	18	17	0
0	9	14	18	18	16	0
0	13	20	22	19	12	0
0	14	23	24	19	11	0
0	14	24	26	20	11	0
0	0	0	0	0	0	0

				10,2236391

- Perhitungan yang dilakukan :

$$(19 \times 0,11067099) + (12 \times 0,11130129) + (0 \times 0,11067099) + (19 \times 0,11130129) \\ +(11 \times 0,11207421) + (0 \times 0,11130129) + (20 \times 0,11067099) + (11 \times 0,11130129) \\ +(0 \times 0,11067099) = 10,2236391$$

berikutnya pindahkan kernel kesudut kiri bawah, kemudian lakukan perhitungan seperti sebelumnya.

0	0	0	0	0	0	0
0	4	7	12	18	17	0
0	9	14	18	18	16	0
0	13	20	22	19	12	0
0	14	23	24	19	11	0
0	14	24	26	20	11	0
0	0	0	0	0	0	0

15,10056066				

- Perhitungan yang dilakukan :

$$(0 \times 0,11067099) + (14 \times 0,11130129) + (23 \times 0,11067099) + (0 \times 0,11130129) + (14 \times 0,11207421) + (24 \times 0,11130129) + (0 \times 0,11067099) + (0 \times 0,11130129) + (0 \times 0,11067099) = 8,34392073$$

Selanjutnya geser kernel satu pixel kekanan, kemudian lakukan perhitungan pada posisi (0,0) dari kernel, seperti yang sebelumnya.

0	0	0	0	0	0	0
0	4	7	12	18	17	0
0	9	14	18	18	16	0
0	13	20	22	19	12	0
0	14	23	24	19	11	0
0	14	24	26	20	11	0
0	0	0	0	0	0	0

	13,90725993			

- Perhitungan yang dilakukan :

$$(14 \times 0,11067099) + (23 \times 0,11130129) + (24 \times 0,11067099) + (14 \times 0,11130129) + (24 \times 0,11207421) + (26 \times 0,11130129) + (0 \times 0,11067099) + (0 \times 0,11130129) + (0 \times 0,11067099) = 13,90725993$$

Selanjutnya geser kernel satu pixel kekanan, kemudian lakukan perhitungan pada posisi (0,0) dari kernel, seperti yang sebelumnya.

0	0	0	0	0	0	0
0	4	7	12	18	17	0
0	9	14	18	18	16	0
0	13	20	22	19	12	0
0	14	23	24	19	11	0
0	14	24	26	20	11	0
0	0	0	0	0	0	0

		15,13059876		

- Perhitungan yang dilakukan :

$$(19 \times 0,11067099) + (11 \times 0,11130129) + (0 \times 0,11067099) + (20 \times 0,11130129) \\ + (1 \times 0,11207421) + (0 \times 0,11130129) + (0 \times 0,11067099) + (0 \times 0,11130129) + \\ (0 \times 0,11067099) = 6,78590511$$

Setelah melakukan perhitungan dengan seluruh nilai *pixel*, didapat hasil seperti tabel berikut ini.

Tabel 4.1 Hasil Perhitungan Konvolusi SSR₁

3,77851134	7,1116749	9,67231767	11,01131007	7,68158325
7,44712461	13,22352909	16,44616428	16,89199974	11,11917462
10,33774683	17,44879143	19,67069037	17,66809989	10,55957649
12,003627	20,00473686	21,89258931	18,22334574	10,2236391
8,34392073	13,90725993	15,13059876	12,34784109	6,78590511

1. Hasil SSR₁

$$SSRn(x,y) = \log l_n(x,y) - \log G_n(x,y)$$

keterangan, $\log l_n(x,y)$ adalah nilai pixel citra awal

Contoh :

- 4 - 3,77851134=0,22148866
- 7 - 7,1116749=-0,1116749
- 12 - 9,67231767=2,32768233
- 18 - 11,01131007=6,98868993
- 17 - 7,68158325=9,31841675
- 9 - 7,44712461=1,55287539
- 14 - 13,22352909=0,77647091
- 18 - 16,44616428=1,55383572
- 18 - 16,89199974=1,10800026
- 16 - 11,11917462=4,88082538
- 13 - 10,33774683=2,66225317
- 20 - 17,44879143=2,55120857

- 22 - 19,67069037=2,32930963
- 19 - 17,66809989=1,33190011
- 12 - 10,55957649=1,44042351
- 14 - 12,003627=1,996373
- 23 - 20,00473686=2,99526314
- 24 - 21,89258931=2,10741069
- 19 - 18,22334574=0,77665426
- 11 - 10,2236391=0,7763609
- 14 - 8,34392073=5,65607927
- 24 - 13,90725993=10,09274007
- 26 - 15,13059876=10,86940124
- 20 - 12,34784109=7,65215891
- 11 - 6,78590511=4,21409489

Maka didapat seluruh hasil perhitungan SSR_1 seperti tabel berikut ini :

Tabel 4.2 Hasil SSR_1

0,22148866	-0,1116749	2,32768233	6,98868993	9,31841675
1,55287539	0,77647091	1,55383572	1,10800026	4,88082538
2,66225317	2,55120857	2,32930963	1,33190011	1,44042351
1,996373	2,99526314	2,10741069	0,77665426	0,7763609
5,65607927	10,09274007	10,86940124	7,65215891	4,21409489

- Misalnya $\sigma = 9,5$

$$\begin{pmatrix} \frac{1}{2\pi(9,5)^2} e^{-\frac{-1^2+1^2}{2(9,5)^2}} & \frac{1}{2\pi(9,5)^2} e^{-\frac{0^2+1^2}{2(9,5)^2}} & \frac{1}{2\pi(9,5)^2} e^{-\frac{1^2+1^2}{2(9,5)^2}} \\ \frac{1}{2\pi(9,5)^2} e^{-\frac{-1^2+0^2}{2(9,5)^2}} & \frac{1}{2\pi(9,5)^2} e^{-\frac{0^2+0^2}{2(9,5)^2}} & \frac{1}{2\pi(9,5)^2} e^{-\frac{1^2+0^2}{2(9,5)^2}} \\ \frac{1}{2\pi(9,5)^2} e^{-\frac{-1^2+1^2}{2(9,5)^2}} & \frac{1}{2\pi(9,5)^2} e^{-\frac{0^2+1^2}{2(9,5)^2}} & \frac{1}{2\pi(9,5)^2} e^{-\frac{1^2+1^2}{2(9,5)^2}} \end{pmatrix}$$

$$\begin{pmatrix} 0.0017449420 & 0.0017546361 & 0.0017449420 \\ 0.0017546361 & 0.0017643841 & 0.0017546361 \\ 0.0017449420 & 0.0017546361 & 0.0017449420 \end{pmatrix}$$

Total keseluruhan : 0.0157626965

Karena total keseluruhan elemen matriks tidak sama dengan 1, maka dilakukan perhitungan ulang dengan rumus :

$$New\alpha = \alpha x \frac{1}{Total}$$

$$New\alpha_{(1,1)} = 0.0017449420 x \frac{1}{0.0157626965} = 0,110700729$$

$$New\alpha_{(1,2)} = 0.0017546361 x \frac{1}{0.0157626965} = 0,111315732$$

$$New\alpha_{(1,3)} = 0.0017449420 x \frac{1}{0.0157626965} = 0,110700729$$

$$New\alpha_{(2,1)} = 0.0017546361 x \frac{1}{0.0157626965} = 0,111315732$$

$$New\alpha_{(2,2)} = 0.0017643841 x \frac{1}{0.0157626965} = 0,111934154$$

$$New\alpha_{(2,3)} = 0.0017546361 x \frac{1}{0.0157626965} = 0,111315732$$

$$New\alpha_{(3,1)} = 0.0017449420 x \frac{1}{0.0157626965} = 0,110700729$$

$$New\alpha_{(3,2)} = 0.0017546361 x \frac{1}{0.0157626965} = 0,111315732$$

$$New\alpha_{(3,3)} = 0.0017449420 x \frac{1}{0.0157626965} = 0,110700729$$

Lakukan perhitungan dengan seluruh elemen matriks, agar mendapatkan hasil seperti berikut :

$$\begin{pmatrix} 0,110700729 & 0,111315732 & 0,110700729 \\ 0,111315732 & 0,111934154 & 0,111315732 \\ 0,110700729 & 0,111315732 & 0,110700729 \end{pmatrix}$$

Selanjutnya dilakukan perhitungan konvolusi antara nilai *pixel* citra awal dengan kernel $G(x,y)$, yaitu $f(x,y) * G(x,y)$ setelah hasil kemudian di tambah,

0	0	0	0	0	0	0			
0	2	2	5	1	4	0			
0	14	29	24	8	3	0	0,11023447	0.11154683	0,11023447
0	10	31	15	9	7	0	0.11154683	0.11287477	0.11154683
0	21	59	38	36	26	0	0,11023447	0.11154683	0,11023447
0	1	55	33	33	25	0			
0	0	0	0	0	0	0			
$F(x,y)$							$G(x,y)$		

Untuk mendapatkan nilai konvolusi maka dilakukan perhitungan dengan nilai matriks $F(x,y)$ dan nilai Gaussian $G(x,y)$, dimulai dari sudut kiri atas matriks.

Lakukan langkah demi langkah perhitungan yang sama untuk mendapatkan hasil konvolusi seperti perhitungan SSR_1 Sebelumnya , sehingga didapat hasil konvolusi pada tabel berikut ini :

Tabel 4.3 Hasil Perhitungan Konvolusi SSR_2

3,77953094	7,11285898	9,67851397	11,02241885	7,69568377
7,45015535	13,22573357	16,44884272	16,8963424	11,12737272
10,34256109	17,45410771	19,67588895	17,67062399	10,56432681
12,00790496	20,01054502	21,89767019	18,22573342	10,22573366
8,35441913	13,92535463	15,1497762	12,36232433	6,79402913

2. Hasil SSR_2

$$SSRn(x,y) = \log l_n(x,y) - \log G_n(x,y)$$

keterangan, $\log l_n(x,y)$ adalah nilai pixel citra awal

Contoh :

- 4 - 3,77953094 = 0,22046906
- 7 - 7,11285898 = -0,11285898
- 12 - 9,67851397 = 2,32148603
- 18 - 11,02241885 = 6,97758115
- 17 - 7,69568377 = 9,30431623

- $9 - 7,45015535 = 1,54984465$
- $14 - 13,22573357 = 0,77426643$
- $18 - 16,44884272 = 1,55115728$
- $18 - 16,8963424 = 1,1036576$
- $16 - 11,12737272 = 4,87262728$
- $13 - 10,34256109 = 2,65743891$
- $20 - 17,45410771 = 2,54589229$
- $22 - 19,67588895 = 2,32411105$
- $19 - 17,67062399 = 1,32937601$
- $12 - 10,56432681 = 1,43567319$
- $14 - 12,00790496 = 1,99209504$
- $23 - 20,01054502 = 2,98945498$
- $24 - 21,89767019 = 2,10232981$
- $19 - 18,22573342 = 0,77426658$
- $11 - 10,22573366 = 0,77426634$
- $14 - 8,35441913 = 5,64558087$
- $24 - 13,92535463 = 10,07464537$
- $26 - 15,1497762 = 10,8502238$
- $20 - 12,36232433 = 7,63767567$
- $11 - 6,79402913 = 4,20597087$

Tabel 4.4 Hasil SSR₂

0,22046906	-0,11285898	2,32148603	6,97758115	9,30431623
1,54984465	0,77426643	1,55115728	1,1036576	4,87262728
2,65743891	2,54589229	2,32411105	1,32937601	1,43567319
1,99209504	2,98945498	2,10232981	0,77426658	0,77426634
5,64558087	10,07464537	10,8502238	7,63767567	4,20597087

- Misalnya $\sigma = 10,5$

$$\begin{pmatrix} \frac{1}{2\pi(10,5)^2} e^{-\frac{-1^2+1^2}{2(10,5)^2}} & \frac{1}{2\pi(10,5)^2} e^{-\frac{0^2+1^2}{2(10,5)^2}} & \frac{1}{2\pi(10,5)^2} e^{-\frac{1^2+1^2}{2(10,5)^2}} \\ \frac{1}{2\pi(10,5)^2} e^{-\frac{-1^2+0^2}{2(10,5)^2}} & \frac{1}{2\pi(10,5)^2} e^{-\frac{0^2+0^2}{2(10,5)^2}} & \frac{1}{2\pi(10,5)^2} e^{-\frac{1^2+0^2}{2(10,5)^2}} \\ \frac{1}{2\pi(10,5)^2} e^{-\frac{-1^2+-1^2}{2(10,5)^2}} & \frac{1}{2\pi(10,5)^2} e^{-\frac{0^2+-1^2}{2(10,5)^2}} & \frac{1}{2\pi(10,5)^2} e^{-\frac{1^2+-1^2}{2(10,5)^2}} \end{pmatrix}$$

$$\begin{pmatrix} 0.0014312732 & 0.0014377790 & 0.0014312732 \\ 0.0014377790 & 0.0014443144 & 0.0014377790 \\ 0.0014312732 & 0.0014377790 & 0.0014312732 \end{pmatrix}$$

Total keseluruhan : 0.0129205232

Karena total keseluruhan elemen matriks tidak sama dengan 1, maka dilakukan perhitungan ulang dengan rumus :

$$New\alpha = \alpha \times \frac{1}{Total}$$

$$New\alpha_{(1,1)} = 0.0014312732 \times \frac{1}{0.0129205232} = 0,110775174$$

$$New\alpha_{(1,2)} = 0.0014377790 \times \frac{1}{0.0129205232} = 0,111278698$$

$$New\alpha_{(1,3)} = 0.0014312732 \times \frac{1}{0.0129205232} = 0,110775174$$

$$New\alpha_{(2,1)} = 0.0014377790 \times \frac{1}{0.0129205232} = 0,111278698$$

$$New\alpha_{(2,2)} = 0.0014443144 \times \frac{1}{0.0129205232} = 0,111784513$$

$$New\alpha_{(2,3)} = 0.0014377790 \times \frac{1}{0.0129205232} = 0,111278698$$

$$New\alpha_{(3,1)} = 0.0014312732 \times \frac{1}{0.0129205232} = 0,110775174$$

$$New\alpha_{(3,2)} = 0.0014377790 \times \frac{1}{0.0129205232} = 0,111278698$$

$$New\alpha_{(3,3)} = 0.0014312732 \times \frac{1}{0.0129205232} = 0,110775174$$

Lakukan perhitungan dengan seluruh elemen matriks, agar mendapatkan hasil seperti berikut :

$$\begin{pmatrix} 0,110775174 & 0,111278698 & 0,110775174 \\ 0,111278698 & 0,111784513 & 0,111278698 \\ 0,110775174 & 0,111278698 & 0,110775174 \end{pmatrix}$$

Selanjutnya dilakukan perhitungan konvolusi nilai pixel citra awal dengan kernel *Gaussian*, yaitu $f(x,y) * G(x,y)$ setelah hasil kemudian di tambah,

0	0	0	0	0	0	0			
0	2	2	5	1	4	0			
0	14	29	24	8	3	0	0,11264953	0,10968526	0,11264953
0	10	31	15	9	7	0	0,10968526	0,11066677	0,10968526
0	21	59	38	36	26	0	0,11264953	0,10968526	0,11264953
0	1	55	33	33	25	0			
0	0	0	0	0	0	0			
$F(x,y)$							$G(x,y)$		

Untuk mendapatkan nilai konvolusi maka dilakukan perhitungan dengan nilai matriks $F(x,y)$ dan nilai *Gaussian* $G(x,y)$, dimulai dari sudut kiri atas matriks.

Lakukan langkah demi langkah perhitungan yang sama untuk mendapatkan hasil *konvolusi* seperti perhitungan SSR_1 dan SSR_2 sebelumnya, sehingga didapat hasil *konvolusi* pada tabel berikut ini :

Tabel 4.5 Hasil Perhitungan Konvolusi SSR_3

3,77472466	7,1067625	9,64925238	10,9772931	7,63832547
7,4377813	13,21746485	16,44079894	16,87657571	11,09390815
10,3231668	17,43289357	19,65524022	17,66313966	10,54155581
11,9901771	19,98824178	21,87758687	18,2174945	10,21744706
8,30831385	13,846856	15,06721396	12,29844351	6,7579186

3. Hasil SSR_3

$$SSRn(x,y) = \log l_n(x,y) - \log G_n(x,y)$$

keterangan, $\log l_n(x,y)$ adalah nilai pixel citra awal

Contoh :

- $4 - 3,77472466 = 0,22527534$
- $7 - 7,1067625 = -0,1067625$
- $12 - 9,64925238 = 2,35074762$
- $18 - 10,9772931 = 7,0227069$
- $17 - 7,63832547 = 9,36167453$
- $9 - 7,4377813 = 1,5622187$
- $14 - 13,21746485 = 0,78253515$
- $18 - 16,44079894 = 1,55920106$
- $18 - 16,87657571 = 1,12342429$
- $16 - 11,09390815 = 4,90609185$
- $13 - 10,3231668 = 2,6768332$
- $20 - 17,43289357 = 2,56710643$
- $22 - 19,65524022 = 2,34475978$
- $19 - 17,66313966 = 1,33686034$
- $12 - 10,54155581 = 1,45844419$
- $14 - 11,9901771 = 2,0098229$
- $23 - 19,98824178 = 3,01175822$
- $24 - 21,87758687 = 2,12241313$
- $19 - 18,2174945 = 0,7825055$
- $11 - 10,21744706 = 0,78255294$
- $14 - 8,30831385 = 5,69168615$
- $24 - 13,846856 = 10,153144$
- $26 - 15,06721396 = 10,93278604$
- $20 - 12,29844351 = 7,70155649$
- $11 - 6,7579186 = 4,2420814$

Tabel 4.6 Hasil SSR_3

0,22527534	-0,1067625	2,35074762	7,0227069	9,36167453
1,5622187	0,78253515	1,55920106	1,12342429	4,90609185
2,6768332	2,56710643	2,34475978	1,33686034	1,45844419
2,0098229	3,01175822	2,12241313	0,7825055	0,78255294
5,69168615	10,153144	10,93278604	7,70155649	4,2420814

b. Metode *Multiscale Retinex*

Setelah melalui tiga tahapan *Singlescale Retinex*, selanjutnya melakukan tahapan *Multiscale Retinex* (MSR), *Multiscale Retinex* adalah generalisasi dari skala tunggal (SSR), dengan Rumus :

$$R_{msri} = \sum_{n=1}^N W_n R_{ni}$$

R_{msri} Merupakan hasil dari *Multiscale Retinex* (MSR) ialah hasil dari jumlah SSR yang masing-masing telah diberi bobot. N ialah jumlah skala yang dipakai. W_n adalah bobot yang diasosiasikan dengan skala ke-n. R_{ni} merupakan output dari SSR yang diasosiasikan dengan skala ke- n. Simbol i menyatakan channel warna, misalnya *Red*(R), *Green*(G), dan *Blue*(B).

Pada tahap melakukan perhitungan agar hasil MSR didapatkan yaitu melakukan penjumlahan nilai *pixel* dari hasil perhitungan SSR_1 SSR_2 dan SSR_3 dan melakukan perkalian pada nilai bobot yaitu 1, sehingga didapat hasil perhitungan *Multiscale Retinex*.

$$MSR = SSR_1 + SSR_2 + SSR_3 \times 1$$

Hasil Perhitungan *Multiscale Retinex* (MSR), seperti tabel dihalaman berikutnya :

Tabel 4.7 Hasil Perhitungan Multiscale Retinex

//0,66723306	-0,33129638	6,99991598	20,98897798	27,98440751
4,66493874	2,33327249	4,66419406	3,33508215	14,65954451
7,99652528	7,66420729	6,99818046	3,99813646	4,33454089
5,99829094	8,99647634	6,33215363	2,33342634	2,33318018
16,99334629	30,32052944	32,65241108	22,99139107	12,66214716

Setelah mendapatkan hasil perhitungan *Multiscale Retinex (MSR)*, nilai *pixel MSR* dinormalisasikan seperti berikut ini :

$$\text{New Value} : \frac{\text{abs}(\text{value} - \text{min})}{(\text{max} - \text{min})} * 255$$

Dari hasil perhitungan MSR mendapatkan nilai :

Nilai terbesar = 32,65241108

Nilai terkecil = -0,33129638

Berdasarkan rentang warna dari nilai terkecil sampai dengan terbesar adalah (0 – 255), maka nilai *pixel* terbesar (32,65241108) merupakan 255 dan nilai minimal (-0,33129638) merupakan 0.

Setelah dilakukan perhitungan normalisasi, hasil yang didapat seperti tabel dibawah ini :

Tabel 4.8 Hasil Normalisasi

7	0	56	164	218
38	20	38	28	115
64	61	56	33	36
48	72	51	20	20
133	236	255	180	100

c. Metode *Median Filter*

Proses *median filter* dilakukan dengan melakukan pergeseran kernel lalu nilai *pixel* yang terdapat pada kernel di urutkan dari nilai terkecil sampai terbesar kemudian didapatlah nilai tengah. Proses pergeseran kernel dilakukan pada seluruh nilai *pixel* yang ada.

Nilai *pixel* yang akan diproses ialah nilai *pixel* citra *Multiscale Retinex* yang telah di normalisasi, kemudia akan dilakukan proses pencarian dengan metode *Median Filter* untuk mendapatkan nilai tengah, seperti langkah - langkah berikut ini :

Tabel 4.9 Nilai *Pixel Multiscale Retinex*

7	0	56	164	218
38	20	38	28	115
64	61	56	33	36
48	72	51	20	20
133	236	255	180	100

$F(x) = 0, 7, 20, 38, (38), 56, 56, 61, 64$

Maka hasil dari pencarian kesembilan kernel/mask adalah 38, kemudian nilai tersebut akan dimasukkan sebagai nilai tengah *pixel* yang baru.

7	0	56	164	218
38	20	38	28	115
64	61	56	33	36
48	72	51	20	20
133	236	255	180	100

$F(x) = 0, 20, 28, 33, (38), 56, 56, 61, 164$

Maka hasil dari pencarian kesembilan kernel/mask adalah 38, kemudian nilai tersebut akan dimasukkan sebagai nilai tengah *pixel* yang baru.

7	0	56	164	218
38	20	38	28	115
64	61	56	33	36
48	72	51	20	20
133	236	255	180	100

$F(x) = 28, 33, 36, 38, (56), 56, 115, 164, 218$

Maka hasil dari pencarian kesembilan kernel/mask adalah 56, kemudian nilai tersebut akan dimasukkan sebagai nilai tengah *pixel* yang baru.

7	0	56	164	218
38	20	38	28	115
64	61	56	33	36
48	72	51	20	20
133	236	255	180	100

$F(x) = 20, 38, 38, 48, (51), 56, 61, 64, 72$

Maka hasil dari pencarian kesembilan kernel/mask adalah 51, kemudian nilai tersebut akan dimasukkan sebagai nilai tengah *pixel* yang baru.

7	0	56	164	218
38	20	38	28	115
64	61	56	33	36
48	72	51	20	20
133	236	255	180	100

$F(x) = 20, 20, 28, 33, (38), 51, 56, 61, 72$

Maka hasil dari pencarian kesembilan kernel/mask adalah 38, kemudian nilai tersebut akan dimasukkan sebagai nilai tengah *pixel* yang baru.

7	0	56	164	218
38	20	38	28	115
64	61	56	33	36
48	72	51	20	20
133	236	255	180	100

$F(x) = 20, 20, 28, 33, (36), 38, 51, 56, 115$

Maka hasil dari pencarian kesembilan kernel/mask adalah 36, kemudian nilai tersebut akan dimasukkan sebagai nilai tengah *pixel* yang baru.

7	0	56	164	218
38	20	38	28	115
64	61	56	33	36
48	72	51	20	20
133	236	255	180	100

$F(x) = 48, 51, 56, 61, (64), 72, 133, 236, 255$

Maka hasil dari pencarian kesembilan kernel/mask adalah 64, kemudian nilai tersebut akan dimasukkan sebagai nilai tengah *pixel* yang baru.

7	0	56	164	218
38	20	38	28	115
64	61	56	33	36
48	72	51	20	20
133	236	255	180	100

$F(x) = 20, 33, 51, 56, (61), 72, 180, 236, 255$

Maka hasil dari pencarian kesembilan kernel/mask adalah 61, kemudian nilai tersebut akan dimasukkan sebagai nilai tengah *pixel* yang baru.

7	0	56	164	218
38	20	38	28	115
64	61	56	33	36
48	72	51	20	20
133	236	255	180	100

$F(x) = 20, 20, 33, 36, (51), 56, 100, 180, 255$

Maka hasil dari pencarian kesembilan kernel/mask adalah 51, kemudian nilai tersebut akan dimasukkan sebagai nilai tengah *pixel* yang baru.

Tabel 4.10 Hasil Output Median Filter

7	0	56	164	218
38	38	38	56	115
64	51	38	36	36
48	64	61	51	20
133	236	255	180	100

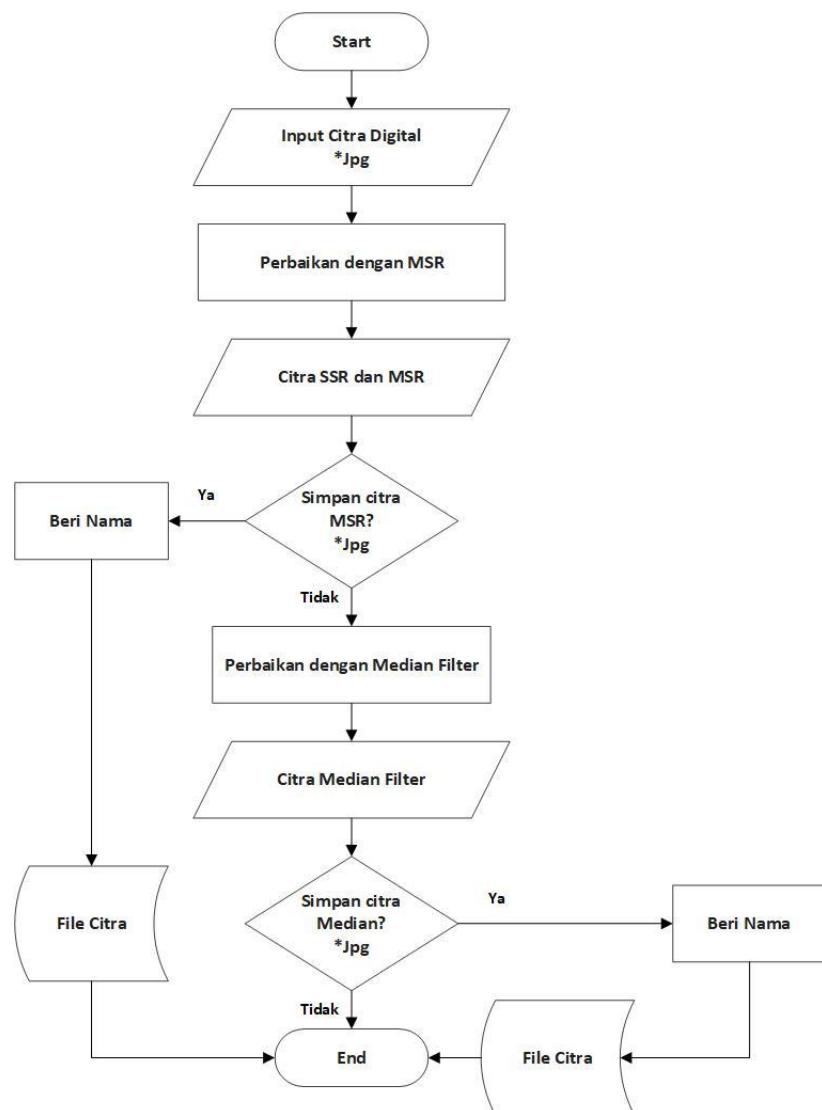
4.1.3 Hasil Analisis Data

Jika sudah menganalisa representasi data, maka kita dapat mengetahui hasil dari masing – masing metode seperti *Singlescale Retinex*, *Multiscale Retinex* beserta nilai konvolusi nya dan *Median Filter*, untuk memperbaiki kualitas citra pada citra digital yang minim intensitas cahaya dan citra digital yang kualitasnya kurang baik.

4.1.4 Perancangan

1. Flowchart

Flowchart merupakan suatu simbol yang menunjukkan tahapan dari suatu program secara mendetail. *Flowchart* berfungsi untuk menjabarkan jalan dari program yang sudah dikerjakan dalam sistem grafis sehingga pengguna lain dapat mengerti dan memahami program yang sudah kita rancang. Rancangan *flowchart* yang akan digunakan untuk memperbaiki kualitas citra digital dengan metode *Multiscale Retinex* dan *Median Filter*, yaitu seperti gambar 4.3 dibawah ini :



Gambar 4.3 Flowchart Sistem Aplikasi

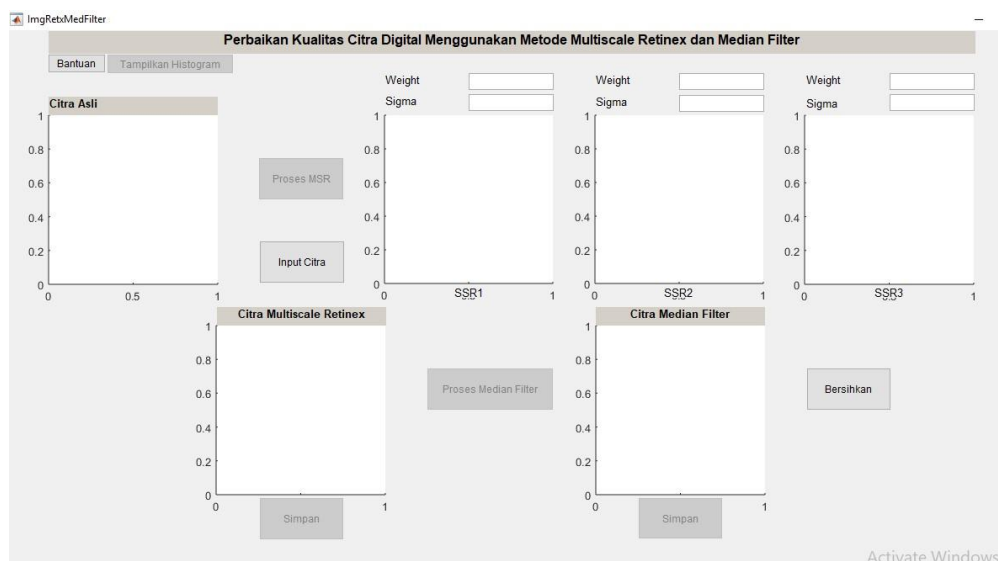
Gambar diatas merupakan alur dari sebuah sistem aplikasi dalam melakukan perbaikan kualitas citra digital dengan menggunakan *Multiscale Retinex* dan *Median Filter*. Dimulai dari menginput citra digital, lalu dilakukan proses perbaikan dengan MSR kemudian akan muncul hasil citra SSR_1 , SSR_2 , SSR_3 dan Citra MSR, jika ingin menyimpan citra hasil *Multiscale Retinex* klik button Simpan kemudian beri nama, jika tidak lanjut melakukan proses perbaikan dengan median filter, maka akan muncul hasil perbaikan citra dengan *median filter*, selanjutnya jika ingin menyimpan klik

button simpan dan beri nama, jika tidak proses perbaikan kualitas citra digital menggunakan *Multiscale Retinex* dan *Median Filter* Selesai.

2. Gambar Tampilan

Tampilan Antar Muka

Form ini merupakan tampilan awal perancangan sistem aplikasi perbaikan kualitas citra digital dengan metode *Multiscale Retinex* dan *Median Filter*.



Gambar 4.4 Tampilan Antar Muka

4.2 Hasil

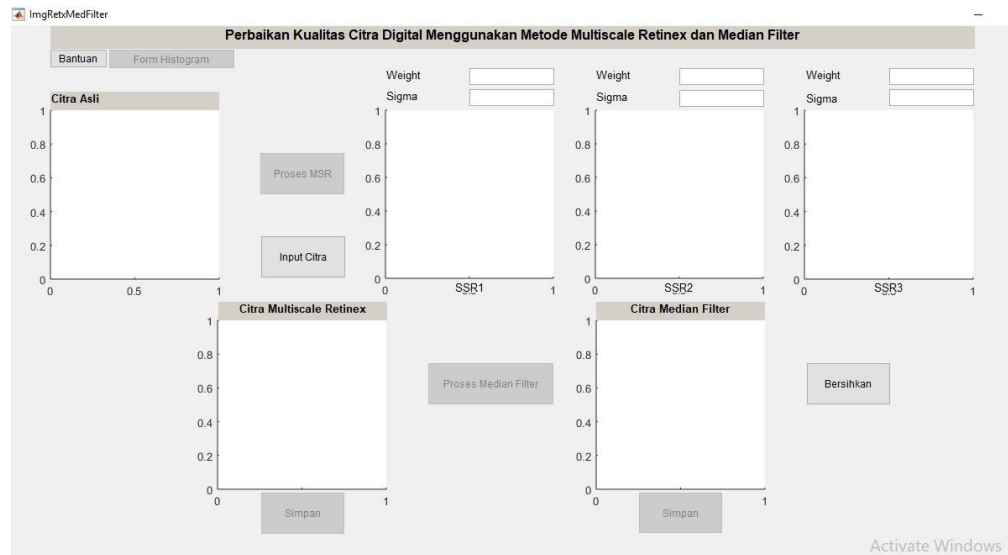
Ada sebagian proses yang harus diketahui terlebih dahulu mengenai hasil pada penelitian ini antara lain pengujian dan penerapan, yaitu :

4.2.1 Pengujian

Pengujian untuk sistem ini dilakukan untuk memperjelas dan mencerahkan citra yang gelap. dilakukan pengujian yang bertujuan agar kita dapat mengetahui hasil ataupun permasalahan pada sistem yang telah dirancang, contoh hasil perbaikan kualitas citra digital dengan metode *Multiscale Retinex* dan *Median Filter* antara lain :

1. Tampilan awal sistem aplikasi

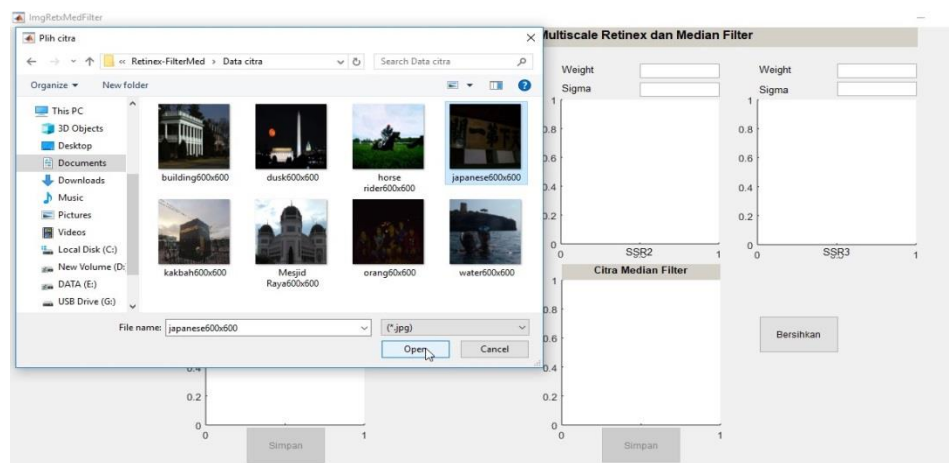
Tampilan pada gambar 4.5 ini merupakan tampilan perbaikan kualitas citra digital menggunakan metode *Multiscale Retinex* dan *Median Filter*.



Gambar 4.5 Tampilan Awal Sistem Aplikasi

2. Penginputan Citra

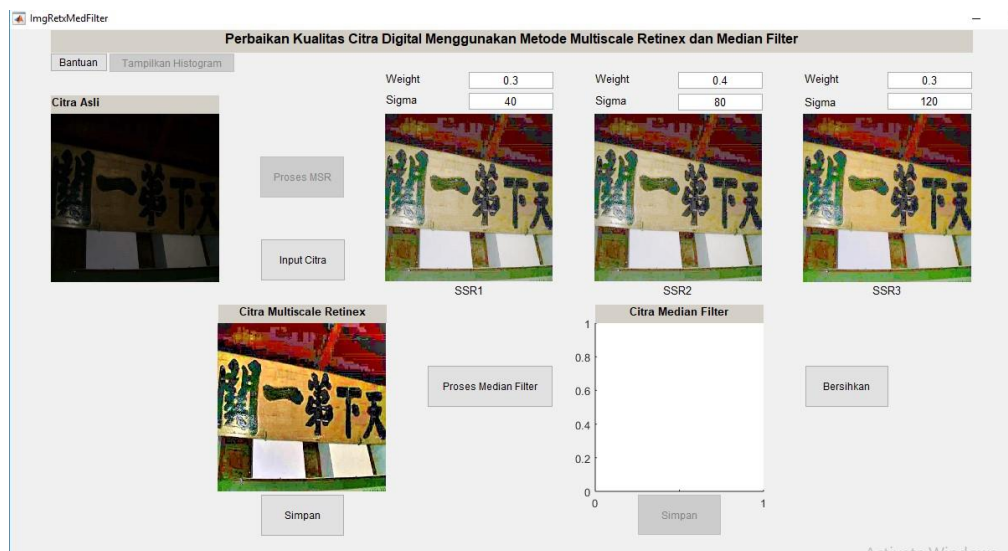
Selanjutnya untuk menjalankannya klik button input citra, untuk mengambil citra digital yang ingin diproses, lalu klik citra dan pilih open, seperti gambar 4.6 berikut ini :



Gambar 4.6 Penginputan Citra

3. Tampilan Setelah Proses *Multiscale Retinex*

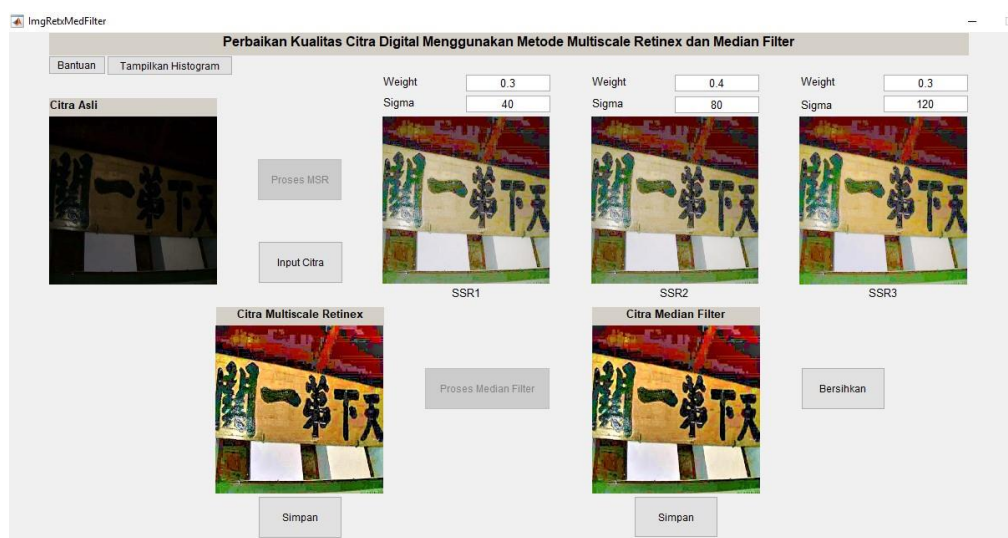
Setelah menginput citra lalu, inputkan weight dan sigma, kemudian klik button Proses MSR, jika ingin menyimpan citra MSR klik button save, sehingga hasil seperti dibawah ini :



Gambar 4.7 Hasil Proses MSR

4. Tampilan Setelah Proses *Median Filter*

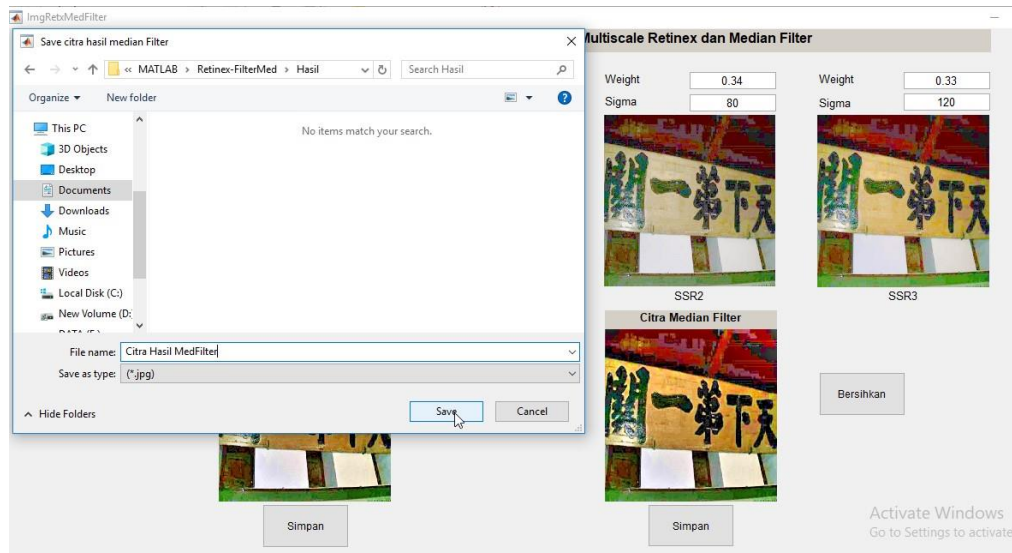
Setelah citra berhasil diterangkan selanjutnya ketahap perbaikan citra digital dengan median filter untuk mngurangi derau citra hasil MSR, klik button proses median filter, hasil seperti dibawah ini:



Gambar 4.8 Hasil Proses Median Filter

5. Tampilan Simpan Citra

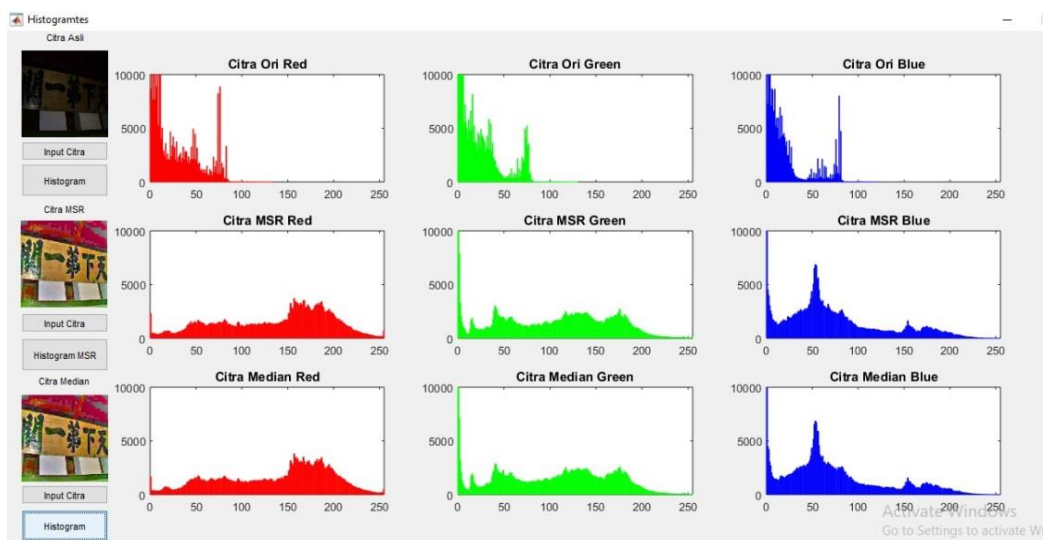
Selanjutnya menyimpan citra hasil median filter, dengan mengklik button save, lalu beri nama dan save.



Gambar 4.9 Simpan Citra

6. Tampilan *Histogram*

Selanjutnya klik button tampilkan *histogram* untuk melihat perbandingan histogram citra asli, citra MSR dan citra *Median Filter*, seperti dibawah ini :




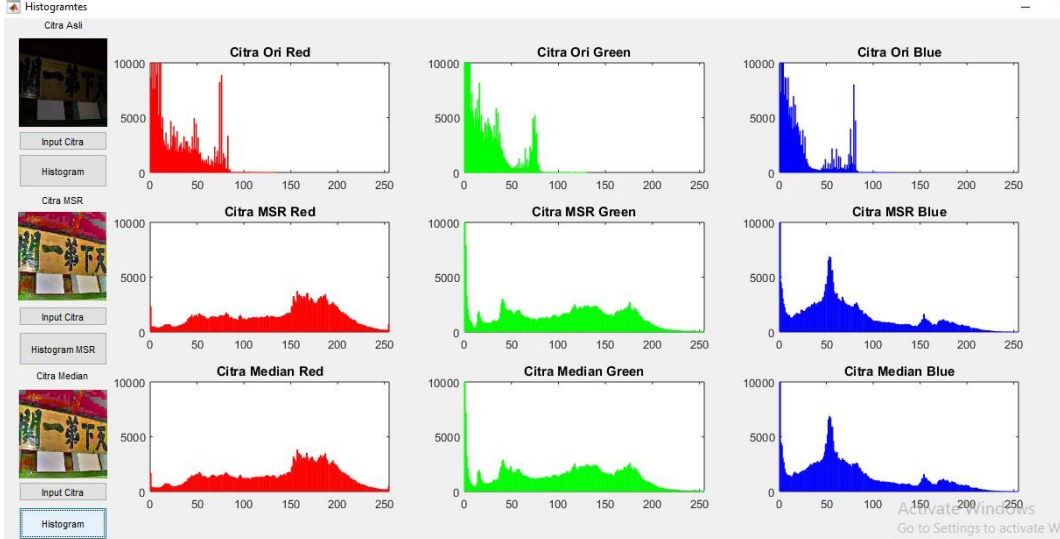


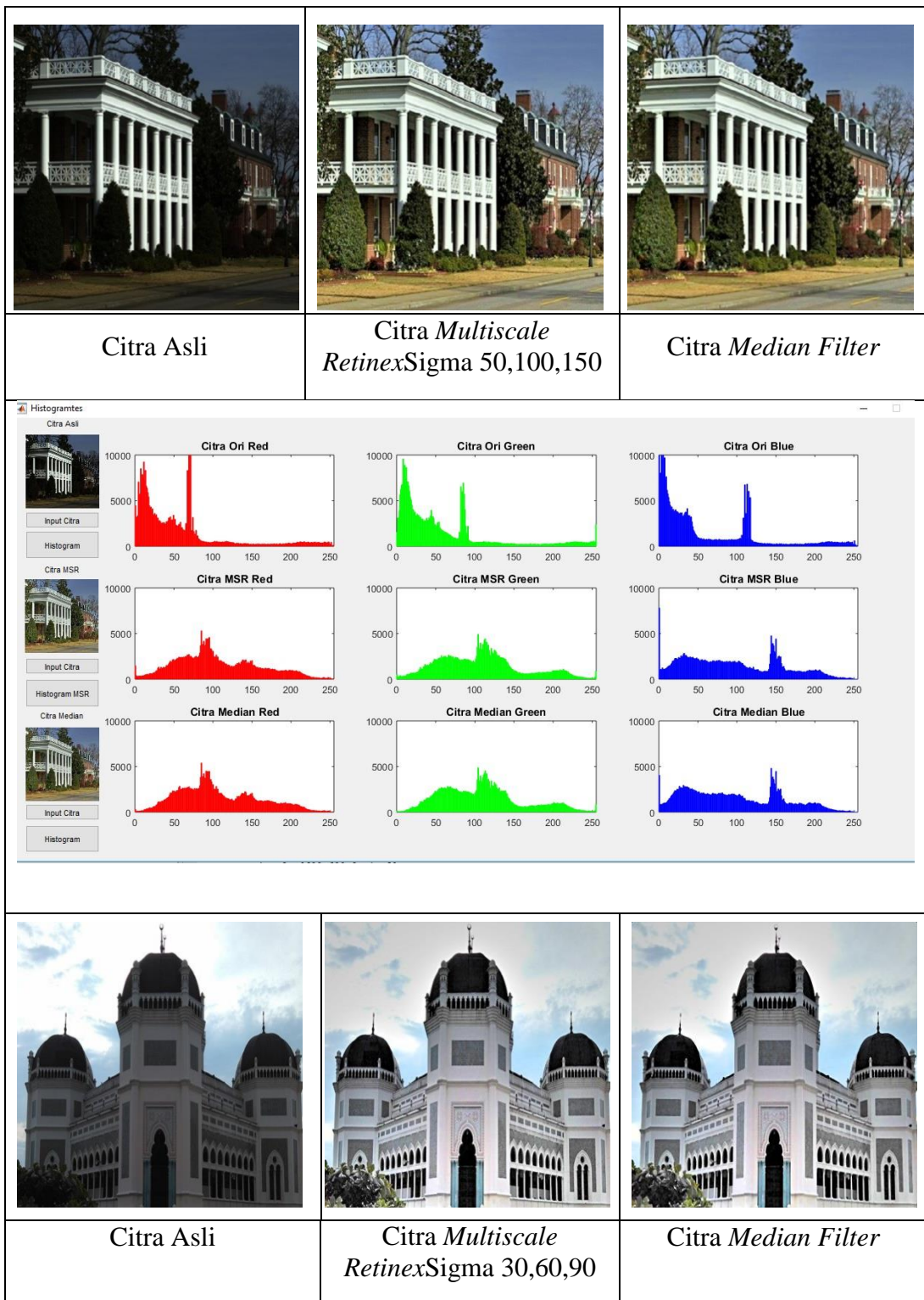
Gambar 4.10 Histogram

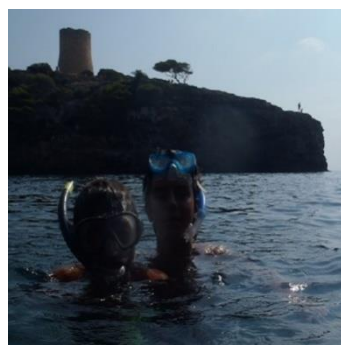
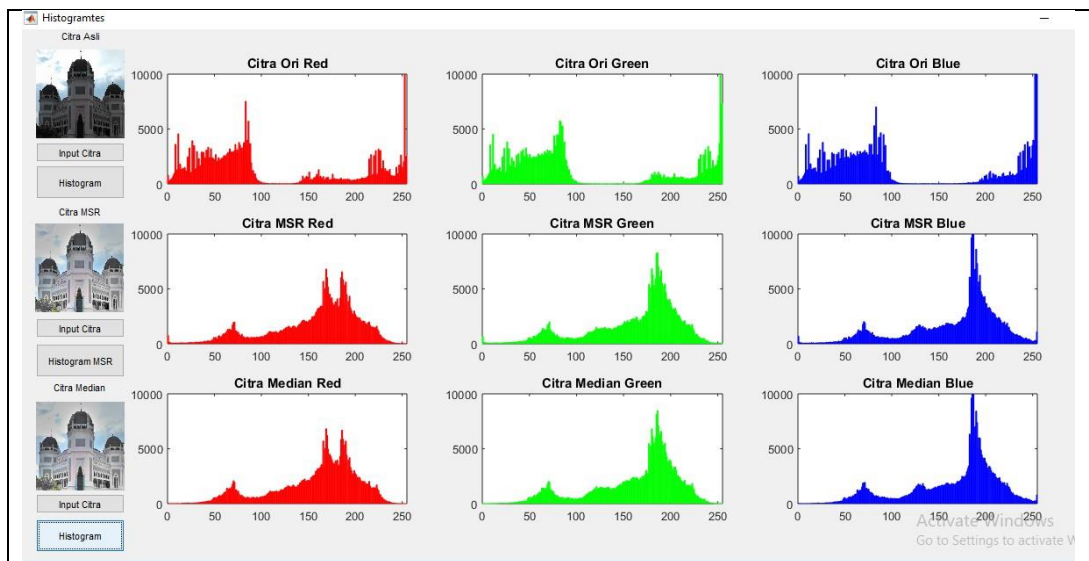
7. Hasil

Setelah citra disimpan dibawah ini merupakan beberapa hasil dari perbaikan kualitas citra digital menggunakan metode *Multiscale Retinex* dan *Median Filter*, beserta *histogram* citra nya.

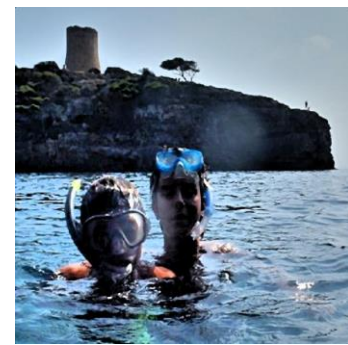
Tabel 4.11 Hasil Citra Digital dan Histogram

Citra Asli	Citra <i>Multiscale Retinex</i>	Citra <i>Median Filter</i>
		
Citra Asli	Citra <i>Multiscale Retinex</i> Sigma 20,40,60	Citra <i>Median Filter</i>
		

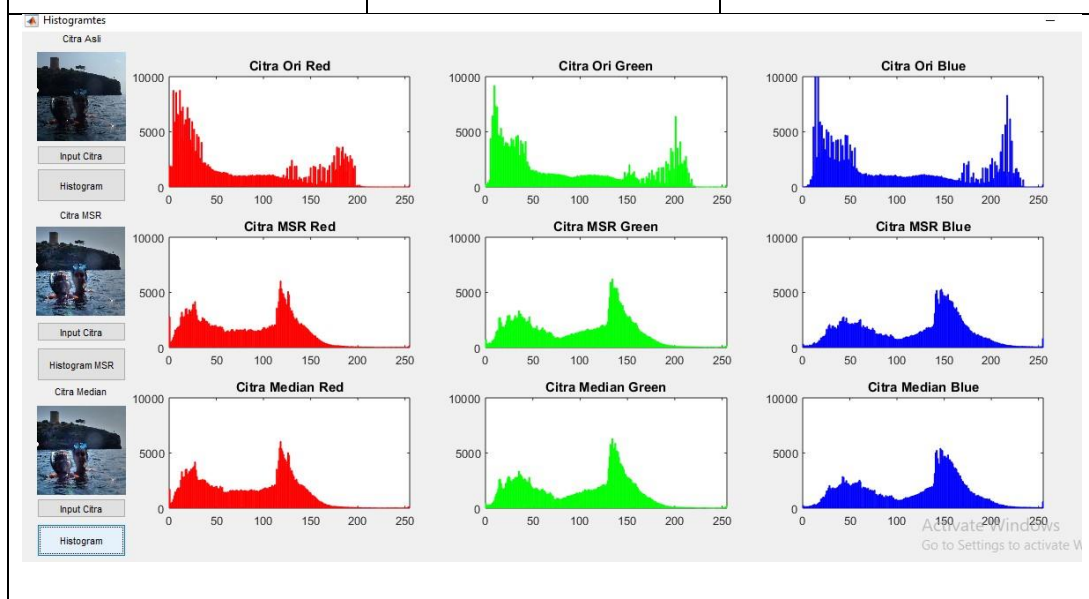





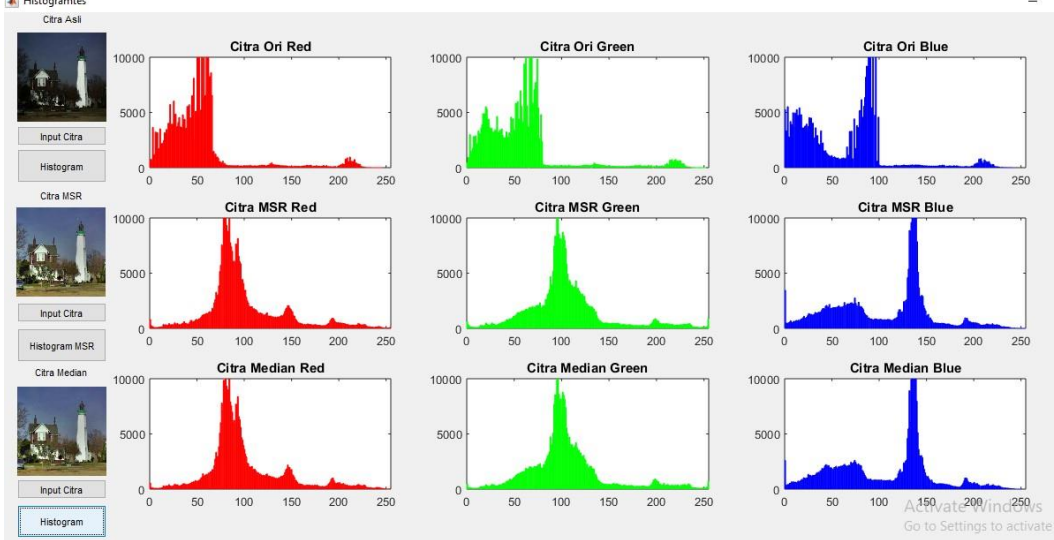





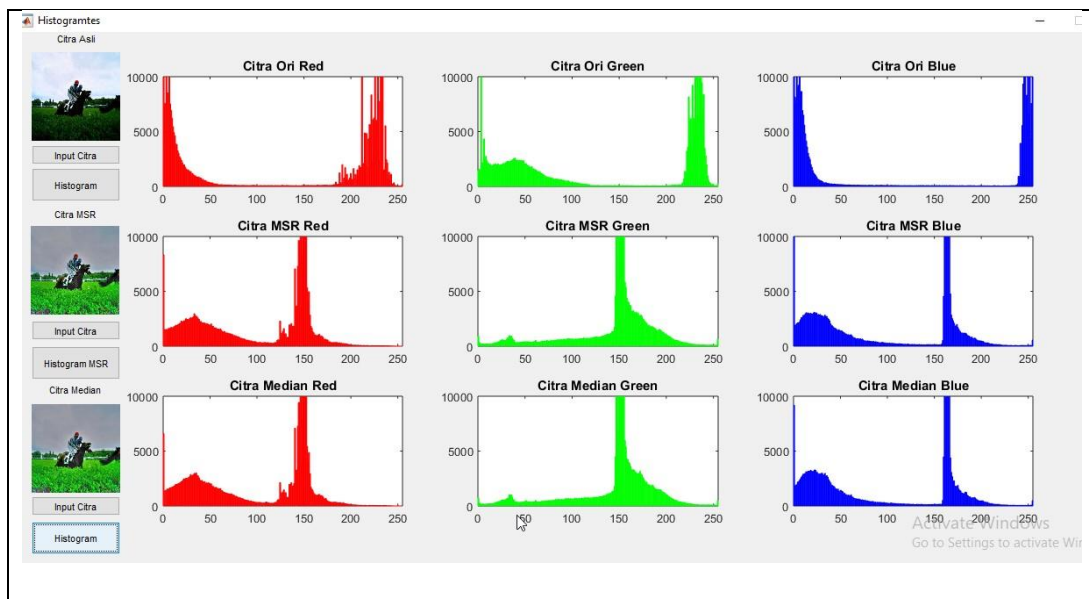
Citra Asli

Citra Multiscale
RetinexSigma 40,80,120

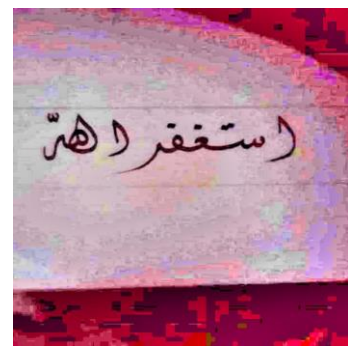
Citra Median Filter



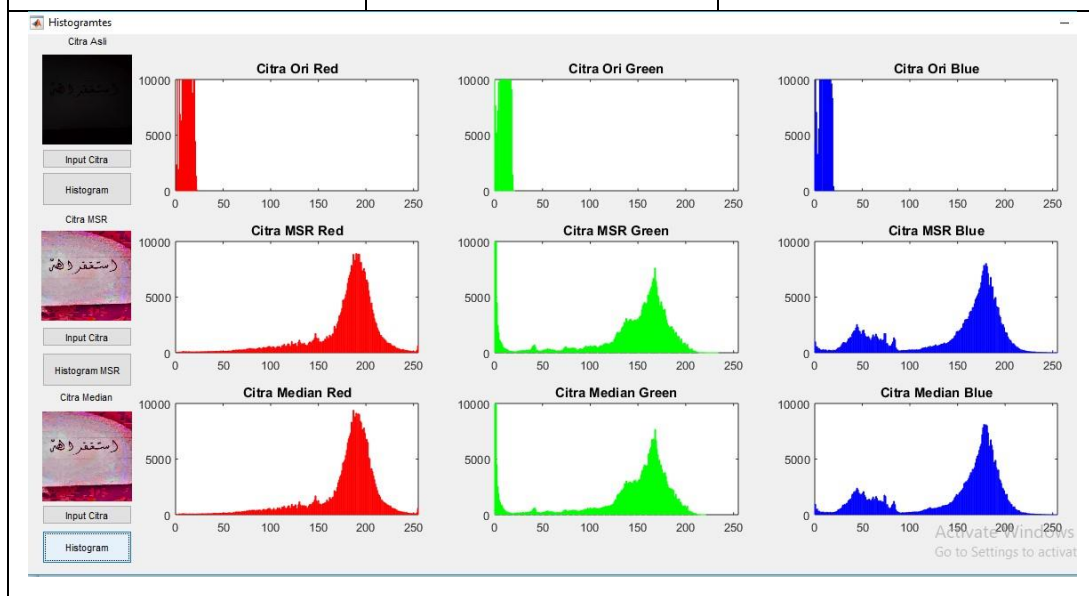
		
Citra Asli	Citra <i>Multiscale Retinex</i> Sigma 50,80,100	Citra <i>Median Filter</i>
		
		
Citra Asli	Citra <i>Multiscale Retinex</i> Sigma 30,40,50	Citra <i>Median Filter</i>

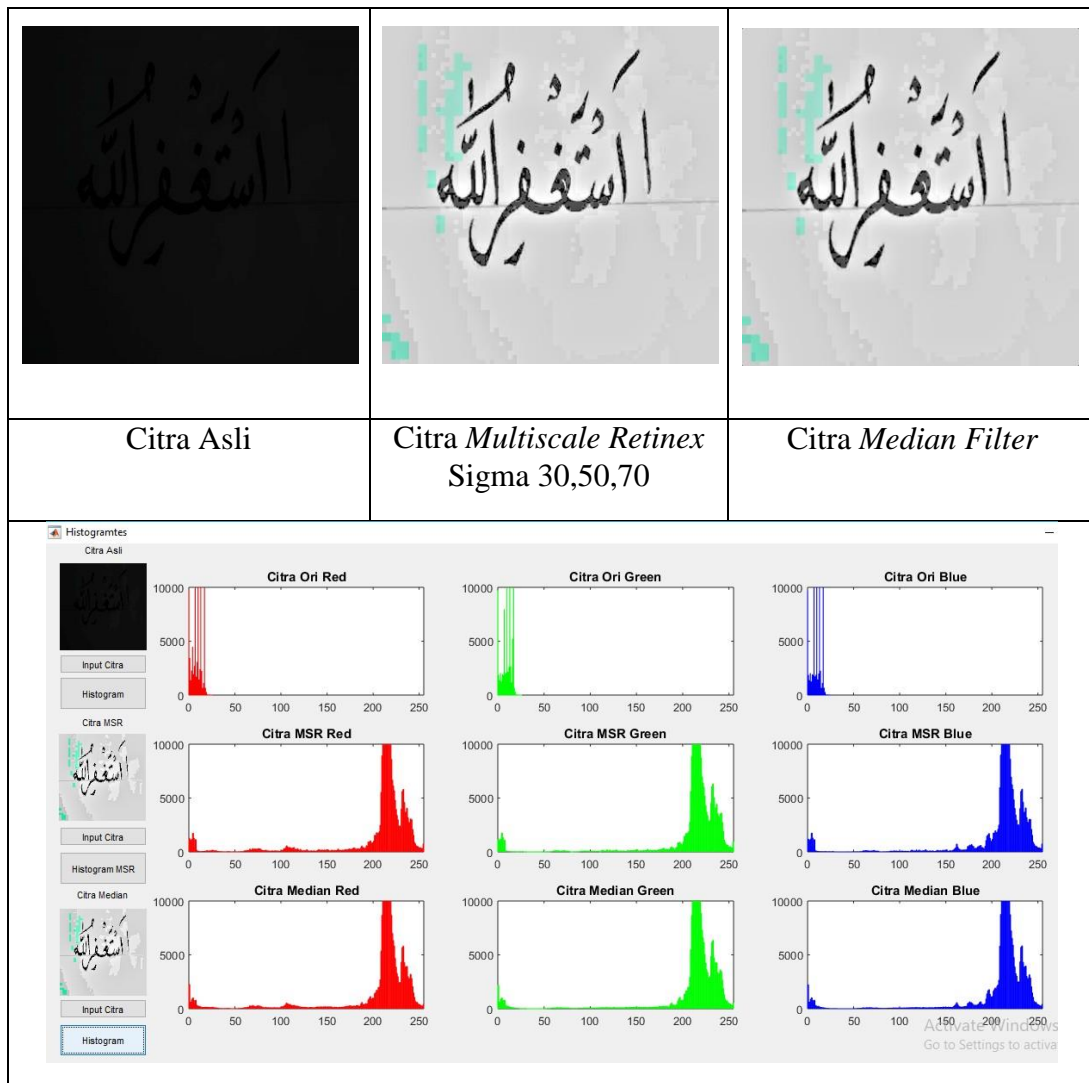


Citra Asli

Citra Multiscale
RetinexSigma 30,40,50

Citra Median Filter





Tabel diatas memperlihatkan citra digital yang sudah diproses dengan *Multiscale Retinex* dan *Median Filter*, dari citra asli yang minim intensitas cahaya kemudian menghasilkan citra yang lebih terang dan jelas, histogram juga menunjukkan perubahan grafik dari citra asli rgb, citra *Multiscale Retinex* rgb dan citra *Median Filter* rgb sehingga terlihat perubahan yang terjadi dari histogram citra dan size ukuran citra juga semakin bertambah setelah diproses menggunakan metode *Multiscale Retinex* dan *Median Filter*.

4.2.2 Penerapan

Penerapan/penggunaan sistem ini untuk memperbaiki kualitas citra digital, dengan menggunakan proses perbaikan kualitas citra (*image enhancement*) menggunakan metode *Multiscale Retinex* dan *Median Filter*, bermanfaat untuk orang-orang yang ingin menerangkan dan menjelaskan citra gelap yang mereka miliki, agar citra tersebut sesuai dengan yang kita harapkan. Sistem ini juga bisa diterapkan untuk pengumpulan barang bukti jika terjadi kasus tindak kriminal, yaitu berupa foto sipelaku yang tertangkap kamera digital dalam keadaan minim intensitas cahaya sehingga kita sulit mengenali wajah sipelaku, dengan sistem ini maka wajah akan terlihat lebih jelas dan terang. Dan juga dapat diterapkan pada citra tulisan yang minim intensitas cahaya sehingga setelah menggunakan sistem ini tulisan tersebut terlihat lebih jelas dan terang.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

1. Dari hasil penerapan metode *Multiscale Retinex* dan *Median filter* yang telah dilakukan, metode *Multiscale Retinex* dapat digunakan dalam mencerahkan citra digital sedangkan metode *Median Filter* dapat digunakan dalam memperjelas citra digital dengan mengurangi *derau/noise* yang terdapat pada citra digital tersebut.
2. Proses *Multiscale Retinex* menciptakan citra yang lebih cerah dibanding dengan citra aslinya dan metode *Median Filter* menciptakan citra yang lebih jelas dibandingkan dengan citra *Multiscale Retinex* sebelum dilakukan proses *median filter*, sebagai perbandingan hasil citra dapat dilihat pada *histogram*, Citra asli Red(R), Green(G) dan Blue(B) *histogram* menunjukkan grafik nilai intensitas *pixel* rata-rata mendekati nol (0) setelah diproses dengan *Multiscale Retinex* dan *Median Fiter* nilai intensitas *pixel* rata-rata menunjukkan perubahan grafik mendekati 250.

5.2 Saran

Beberapa saran dalam penelitian ini antara lain sebagai berikut :

1. Sistem ini hanya menggunakan format gambar *.Jpg, dapat dikembangkan lebih lanjut dengan format gambar lainnya.
2. Menggunakan metode pengurangan noise yang lain, seperti *Mean filter* dan *Gaussian filter*
3. Melakukan perbandingan antara Metode *Multiscale Retinex* dengan metode-metode *image enhancement* lainnya, sehingga mengetahui metode apa yang paling bagus untuk *image enhancement*.

DAFTAR PUSTAKA

- Affandi Muhajir, (2018). *Teknologi Informasi & Komunikasi Dalam Pendidikan*. Kuningan : YNHW
- Bangalore, T. (n.d.). *Color Image Enhancement using Multiscale Retinex with Modified Color Restoration Technique*.
- Cahyono, B. (n.d.). (*Matlab*) *Dalam Pembelajaran*. 1, 45–62.
- Fakhri, J. (2010). Sains Dan Teknologi Dalam Al-Qur'an Dan Implikasinya Dalam Pembelajaran. *Ta'dib*, 15(01), 121–142.
- Furqan, M., & Siregar, Y. K. (2020). *Perbandingan Algoritma Contraharmonic Mean Filter Dan Arithmetic Mean Filter untuk Mereduksi Exponential Noise*. 5(2), 107–115.
- Gunadarma, U. (2005). *Pendahuluan Pengantar Pengolahan Citra*.
- Harahap, B. (2018). *Implementasi Metode Retinex Uuntuk Meningkatkan Kualitas*. 2, 193–200.
- Hasugian, A. H., & Zufira, I. (2018). Perancangan Sistem Restorasi Citra Dengan Metode Image Inpainting. *Jurnal Ilmu Komputer Dan Informatika*, 03(November), 31–45.
- Hong-wei, Z., & Jiu-gen, Z. (2018). *Color Image Enhancement Using Multiscale Retinex Based on Particle Swarm Optimization Method Color Image Enhancement Using Multiscale Retinex Based on Particle Swarm Optimization Method*. 6–12.
- Kusumanto, R. (2011). Technogenic activity of man and local sources of environmental pollution. *Studies in Environmental Science*, 17(C), 329–332.
- Li, B., Wang, S., & Geng, Y. (2011). *Image Enhancement Based On Retinex And Lighhtness Decomposition* Bo Li, Shuhang Wang, Yanbing Geng *Digital Multimedia Laboratory, Computer Science and Engineering, Beihang University, Beijing 100191*. 3478–3481.
- Maulana, I., Andono, P. N., Studi, P., Informatika, T., Komputer, F. I., & Nuswantoro, U. D. (2016). *Analisa Perbandingan Adaptif Median Filter Dan Median Filter Dalam Reduksi Noise Salt & Pepper*. 2(2), 157–166.

- Munawaroh, S., & Sutanto, F. A. (2010). *Pengolah Citra Digital untuk Identifikasi Uang Kertas*. XV(1), 34–40.
- Ngafifi, M. (2014). Kemajuan Teknologi Dan Pola Hidup Manusia Dalam Perspektif Sosial Budaya. *Jurnal Pembangunan Pendidikan: Fondasi Dan Aplikasi*, 2(1), 33–47.
- Nur, S., Capah, A., Nasution, S. D., & Hondro, R. K. (2018). *Penerapan Metode Median Filter Untuk Mereduksi*. 6, 274–277.
- Parihar, A. S., & Singh, K. (2019). A study on Retinex based method for image enhancement. *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, April, 619–624.
- Prasetyo Dwi. (2018). *Peningkatan Ekstraksi Fitur Berbasis Scale Invariant Feature Transform Menggunakan Metode Multiscale Retinex Untuk Meningkatkan Jumlah Keypoint*. Vol.2 N0.2
- Purba, B. (2017). *Aplikasi Perbaikan Kualitas Citra Hasil Penginderaan Jauh (Remote Sensing) Dengan Metode Contrast Stretching*. VI(2), 26–36.
- Putra Darma. (2010). *Pengolahan Citra Digital*. Yogyakarta : Andi
- Rahman, Z., Woodell, G. A., & Jobson, D. J. (2020). *A Comparison of the Multiscale Retinex With Other Image Enhancement Techniques*. 2, 1–6.
- Sholihin, R. A., Purwoto, B. H., Elektro, J. T., Teknik, F., Surakarta, U. M., & Enhancement, I. (2013). *Perbaikan Citra Dengan Menggunakan Median Filter dan Metode Histogram Equalization*. 14(02).
- Sulistiyanti, S. R. (2016). *Pengolahan Citra*.
- Suprpto, (2008). *Bahasa Pemrograman*. Jakarta : Departemen Pendidikan Nasional.
- Supriadi, D., Novantara, P., Ilmu, F., & Universitas, K. (2017). *Implementasi Metode Multiscale Retinex Untuk Image Enhancement*.
- Sutoyo, T., Mulyanto, E., Suhartono, V., Nurhayati, O., D., & Wijanarto. (2009) *Teori Pengolahan Citra Digital* . Yogyakarta : Penerbit ANDI.
- Syamsuddin, S. (2019). Aplikasi Peningkatan Kualitas Citra Menggunakan Metode Median Filtering Untuk Menghilangkan Noise. *SISITI: Seminar Ilmiah Sistem Informasi Dan Teknologi Informasi*, 8(1), 227–236.

Tarigan, A. K., Nasution, S. D., & Karim, A. (2016). *Aplikasi Pembelajaran Citra Dengan Menggunakan Metode Computer Assisted Instruction (Cai)*. 3(4), 1–4.

LAMPIRAN – LAMPIRAN

Lampiran 1

Tools Matlab Gui untuk Perbaikan Kualitas Citra Digital Menggunakan Metode
Multiscale Retinex dan Median Filter :

Tools	Fungsi
axes1	Menampilkan gambar inputan
axes3	Menampilkan gambar SSR1
axes4	Menampilkan gambar SSR2
axes5	Menampilkan gambar SSR3
axes6	Menampilkan gambar hasil MSR
axes7	Menampilkan gambar hasil Median
pushbutton1	Mengambil gambar
pushbutton2	Memproses MSR
pushbutton3	Menyimpan gambar MSR
pushbutton4	Memproses Median Filter
pushbutton5	Menyimpan gambar Median filter
pushbutton6	Membersihkan program
pushbutton7	Menampilkan form bantuan
pushbutton 8	Menampilkan Histogram

Lampiran 2

Kode program Matlab yang digunakan untuk Perbaikan Kualitas Citra Digital

Menggunakan Metode *Multiscale Retinex* dan *Median Filter* :

```
%Buttton Input Citra
function btnLoad_Callback(hObject, eventdata, handles)
global Citra;
[filename, pathname, filterindex] =
uigetfile({'*.jpg'}, 'Plih citra')
handles.myImage = strcat(pathname, filename);
Citra=imread(handles.myImage);
gg=strcat(pathname, filename);
axes(handles.axes1);
imshow(Citra);
set (handles.W1, 'string');
set (handles.W2, 'string');
set (handles.W3, 'string');

set (handles.S1, 'string');
set (handles.S2, 'string');
set (handles.S3, 'string');
set (handles.btnMSR, 'enable', 'on');

%Button Multiscale Retinex
function btnMSR_Callback(hObject, eventdata, handles)
global Citra;
global MSR;
global I;
I=Citra;
Ir=I(:, :, 1);
Ig=I(:, :, 2);
Ib=I(:, :, 3);

G = 192;
b = -30;
alpha = 125;
beta = 46;
Ir_double=double(Ir);
Ig_double=double(Ig);
Ib_double=double(Ib);

sigma_1=str2num(get(handles.S1, 'string'));
sigma_2=str2num(get(handles.S2, 'string'));
sigma_3=str2num(get(handles.S3, 'string'));
weight_1=str2num(get(handles.W1, 'string'));
weight_2=str2num(get(handles.W2, 'string'));
weight_3=str2num(get(handles.W3, 'string'));
```

```

%Perhitungan Gaussian
[x, y]=meshgrid((- (size(Ir,2)-1)/2):(size(Ir,2)/2), (-
(size(Ir,1)-1)/2):(size(Ir,1)/2));
gauss_1=exp(-(x.^2+y.^2)/(2*sigma_1*sigma_1));
Gauss_1=gauss_1/sum(gauss_1(:));
gauss_2=exp(-(x.^2+y.^2)/(2*sigma_2*sigma_2));
Gauss_2=gauss_2/sum(gauss_2(:));
gauss_3=exp(-(x.^2+y.^2)/(2*sigma_3*sigma_3));
Gauss_3=gauss_3/sum(gauss_3(:));

Ir_log=log(Ir_double+1);
f_Ir=fft2(Ir_double);

fgauss=fft2(Gauss_1,size(Ir,1),size(Ir,2));
fgauss=fftshift(fgauss);
Rr=ifft2(fgauss.*f_Ir);
min1=min(min(Rr));
Rr_log= log(Rr - min1+1);
Rr1=Ir_log-Rr_log;

fgauss=fft2(Gauss_2,size(Ir,1),size(Ir,2));
fgauss=fftshift(fgauss);
Rr= ifft2(fgauss.*f_Ir);
min1=min(min(Rr));
Rr_log= log(Rr - min1+1);
Rr2=Ir_log-Rr_log;

fgauss=fft2(Gauss_3,size(Ir,1),size(Ir,2));
fgauss=fftshift(fgauss);
Rr= ifft2(fgauss.*f_Ir);
min1=min(min(Rr));
Rr_log= log(Rr - min1+1);
Rr3=Ir_log-Rr_log;

Rr=(weight_1*Rr1)+(weight_2*Rr2)+(weight_3*Rr3);    %Weighted
summation
MSR1 = Rr;
SSR1 = Rr2;

CRr = beta*(log(alpha*Ir_double+1)-
log(Ir_double+Ig_double+Ib_double+1));

%SSR
min1 = min(min(SSR1));
max1 = max(max(SSR1));
SSR1 = uint8(255*(SSR1-min1)/(max1-min1));

min1 = min(min(MSR1));
max1 = max(max(MSR1));
MSR1 = uint8(255*(MSR1-min1)/(max1-min1));

```

```

%MSR
Rr = G*(CRr.*Rr+b);
min1 = min(min(Rr));
max1 = max(max(Rr));
Rr_final = uint8(255*(Rr-min1)/(max1-min1));

Ig_log=log(Ig_double+1);
f_Ig=fft2(Ig_double);
fgauss=fft2(Gauss_1,size(Ig,1),size(Ig,2));
fgauss=fftshift(fgauss);
Rg= ifft2(fgauss.*f_Ig);
min2=min(min(Rg));
Rg_log= log(Rg-min2+1);
Rg1=Ig_log-Rg_log;

fgauss=fft2(Gauss_2,size(Ig,1),size(Ig,2));
fgauss=fftshift(fgauss);
Rg= ifft2(fgauss.*f_Ig);
min2=min(min(Rg));
Rg_log= log(Rg-min2+1);
Rg2=Ig_log-Rg_log;

fgauss=fft2(Gauss_3,size(Ig,1),size(Ig,2));
fgauss=fftshift(fgauss);
Rg= ifft2(fgauss.*f_Ig);
min2=min(min(Rg));
Rg_log= log(Rg-min2+1);
Rg3=Ig_log-Rg_log;

Rg=(weight_1*Rg1)+(weight_2*Rg2)+(weight_3*Rg3);
%Perhitungan Weight
SSR2 = Rg2;
MSR2 = Rg;
CRg = beta*(log(alpha*Ig_double+1)-
log(Ir_double+Ig_double+Ib_double+1));

%SSR:
min2 = min(min(SSR2));
max2 = max(max(SSR2));
SSR2 = uint8(255*(SSR2-min2)/(max2-min2));

%MSR
min2 = min(min(MSR2));
max2 = max(max(MSR2));
MSR2 = uint8(255*(MSR2-min2)/(max2-min2));

Rg = G*(CRg.*Rg+b);
min2 = min(min(Rg));
max2 = max(max(Rg));
Rg_final = uint8(255*(Rg-min2)/(max2-min2));

```

```

Ib_log=log(Ib_double+1);
f_Ib=fft2(Ib_double);

fgauss=fft2(Gauss_1,size(Ib,1),size(Ib,2));
fgauss=fftshift(fgauss);
Rb= ifft2(fgauss.*f_Ib);
min3=min(min(Rb));
Rb_log= log(Rb-min3+1);
Rb1=Ib_log-Rb_log;

fgauss=fft2(Gauss_2,size(Ib,1),size(Ib,2));
fgauss=fftshift(fgauss);
Rb= ifft2(fgauss.*f_Ib);
min3=min(min(Rb));
Rb_log= log(Rb-min3+1);
Rb2=Ib_log-Rb_log;

fgauss=fft2(Gauss_3,size(Ib,1),size(Ib,2));
fgauss=fftshift(fgauss);
Rb= ifft2(fgauss.*f_Ib);
min3=min(min(Rb));
Rb_log= log(Rb-min3+1);
Rb3=Ib_log-Rb_log;

Rb=(weight_1*Rb1)+(weight_2*Rb2)+(weight_3*Rb3);

CRb = beta*(log(alpha*Ib_double+1)-
log(Ir_double+Ig_double+Ib_double+1));
SSR3 = Rb2;
MSR3 = Rb;
%SSR:
min3 = min(min(SSR3));
max3 = max(max(SSR3));
SSR3 = uint8(255*(SSR3-min3)/(max3-min3));

%MSR
min3 = min(min(MSR3));
max3 = max(max(MSR3));
MSR3 = uint8(255*(MSR3-min3)/(max3-min3));

Rb = G*(CRb.*Rb+b);
min3 = min(min(Rb));
max3 = max(max(Rb));
Rb_final = uint8(255*(Rb-min3)/(max3-min3));
Int = (Ir_double + Ig_double + Ib_double) / 3.0;
Int_log = log(Int+1);
f_Int=fft2(Int_log);

fgauss=fft2(Gauss_1,size(Int,1),size(Int,2));
fgauss=fftshift(fgauss);
RInt=ifft2(fgauss.*f_Int);

```

```

min1=min(min(RInt));
RInt_log= RInt - min1+1;
RInt1=Int_log-RInt_log;

fgauss=fft2(Gauss_2,size(Int,1),size(Int,2));
fgauss=fftshift(fgauss);
RInt= ifft2(fgauss.*f_Int);
min1=min(min(RInt));
RInt_log= RInt - min1+1;
RInt2=Int_log-RInt_log;

fgauss=fft2(Gauss_3,size(Int,1),size(Int,2));
fgauss=fftshift(fgauss);
RInt= ifft2(fgauss.*f_Int);
min1=min(min(RInt));
RInt_log= RInt - min1+1;
RInt3=Int_log-RInt_log;

RInt=(weight_1*RInt1)+(weight_2*RInt2)+(weight_3*RInt3);

minInt = min(min(RInt));
maxInt = max(max(RInt));
Int1 = uint8(255*(RInt-minInt)/(maxInt-minInt));

MSRr = zeros(size(I, 1), size(I, 2));
MSRg = zeros(size(I, 1), size(I, 2));
MSRb = zeros(size(I, 1), size(I, 2));

for ii = 1 : size(I, 1)
for jj = 1 : size(I, 2)
    C = max(Ig_double(ii, jj), Ib_double(ii, jj));
    B = max(Ir_double(ii, jj), C);
    A = min(255.0 / B, Int1(ii, jj) / Int(ii, jj));
    MSRr(ii, jj) = A * Ir_double(ii, jj);
    MSRg(ii, jj) = A * Ig_double(ii, jj);
    MSRb(ii, jj) = A * Ib_double(ii, jj);
end
end

minInt = min(min(MSRr));
maxInt = max(max(MSRr));
MSRr = uint8(255*(MSRr-minInt)/(maxInt-minInt));

minInt = min(min(MSRg));
maxInt = max(max(MSRg));
MSRg = uint8(255*(MSRg-minInt)/(maxInt-minInt));

minInt = min(min(MSRb));
maxInt = max(max(MSRb));
MSRb = uint8(255*(MSRb-minInt)/(maxInt-minInt));

```



```

SSR1 = cat(3,SSR1,SSR2,SSR3);
SSR2 = cat(3,MSR1,MSR2,MSR3);
SSR3=cat(3,Rr_final,Rg_final,Rb_final);
MSR = cat(3, MSRr, MSRg, MSRb);

if (sigma_1 > 0) & (sigma_2 > 0) & (sigma_3 > 0) & (weight_1
> 0) & (weight_2 > 0) & (weight_3 >0)
if (weight_1 + weight_2 + weight_3) == 1
    sigma_1=str2num(get(handles.S1,'string'));
    sigma_2=str2num(get(handles.S2,'string'));
    sigma_3=str2num(get(handles.S3,'string'));

    weight_1=str2num(get(handles.W1,'string'));
    weight_2=str2num(get(handles.W2,'string'));
    weight_3=str2num(get(handles.W3,'string'));

    axes(handles.axes3);
    imshow(SSR1);

    axes(handles.axes4);
    imshow(SSR2);

    axes(handles.axes5);
    imshow(SSR3);
    Im=Citra;
    L = max(Im, [], 3);
    %MSRetinex2(im, sigmaS, saturatedpix, precision)
    ret2 = MSRetinex2(mat2gray(L), [sigma_1,
sigma_2,sigma_3], [weight_1 weight_2], 8);
    %ret2 = MSRetinex2(mat2gray(L), sigmaS,
saturatedpix, precision);
    Ihsv = rgb2hsv(Im);
    Ihsv(:, :, 3) = mat2gray(ret2);
    R2 = hsv2rgb(Ihsv);
    MSR=R2;
    axes(handles.axes6);
    imshow(MSR);

    set (handles.btnMFilter,'enable','on');
    set (handles.btnSaveMRS,'enable','on');
    set (handles.btnMSR,'enable','off');
else
    msgbox('Total Weight harus = 1');
end
else
    msgbox('Isi Parameter Sigma dan Weight dengan benar');
end

```

```

%Button Simpan Citra MSR
function btnSaveMRS_Callback(hObject, eventdata, handles)
global MSR;
[namafile,direktori]=uiputfile({'*.jpg';'*.bmp'}, ' Save
citra hasil');
if isequal (namafile,0)
    errordlg('Error ..!', 'Nama file citra tidak ada');
return;
else
    imwrite(MSR, strcat(direktori, namafile));
    msgbox('Berhasil Menyimpan Citra');
end
set (handles.btnSaveMRS, 'enable', 'off');

%Button Median Filter
function btnMFilter_Callback(hObject, eventdata, handles)
global MSR;
global MSRFilter;
img = MSR;
R = img (:,:, 1);
G = img (:,:, 2);
B = img (:,:, 3);
medfiling (:,:, 1) = medfilt2 (R);
medfiling (:,:, 2) = medfilt2 (G);
medfiling (:,:, 3) = medfilt2 (B);
MSRFilter=medfiling;
axes(handles.axes7);
imshow(MSRFilter);

set (handles.btnMFilter, 'enable', 'off');
set (handles.btnSaveMedFil, 'enable', 'on');
set (handles.btnHisto, 'enable', 'on');

%Button Simpan Citra Median Filter
function btnSaveMedFil_Callback(hObject, eventdata, handles)
global MSRFilter;
[namafile,direktori]=uiputfile({'*.jpg'}, ' Save citra hasil
median Filter');
if isequal (namafile,0)
    errordlg('Error ..!', 'Nama file citra tidak ada');
return;
else
    imwrite(MSRFilter, strcat(direktori, namafile));
    msgbox('Berhasil Menyimpan Citra');
end
set (handles.btnSaveMedFil, 'enable', 'off');

%Button Bersihkan
function btnClear

```

```

set(handles.W1,'string','');
set(handles.W2,'string','');
set(handles.W3,'string','');
set(handles.S1,'string','');
set(handles.S2,'string','');
set(handles.S3,'string','');
axes(handles.axes1);
cla('reset');
axes(handles.axes3);
cla('reset');
axes(handles.axes4);
cla('reset');
axes(handles.axes5);
cla('reset');
axes(handles.axes6);
cla('reset');
axes(handles.axes7);
cla('reset');
clc;
clc;
clc;
clc;
clear all;

%Button Menampilkan Histogram
% --- button input citra msr
function
pushbutton6[nama_file,nama_path]=uigetfile({'*.jpg','File
JPG (*.jpg)'},'Buka File Citra')
handles.citra2 = imread(fullfile(nama_path,nama_file))
axes(handles.axes2)
imshow(handles.citra2)

% --- Histogram Msr
function pushbutton3
Img = handles.citra2;
R = Img(:,:,1);
G = Img(:,:,2);
B = Img(:,:,3);

Merah = cat(3,R,G*0,B*0);
Hijau = cat(3,R*0,G,B*0);
Biru = cat(3,R*0,G*0,B);

subplot(3,3,4);
his(R(:),256,'FC','r','EC','r')
set(gca,'XLim',[0 255])
set(gca,'YLim',[0 10000])
title('Citra MSR Red')
subplot(3,3,5);
his(G(:),256,'FC','g','EC','g')

```

```

set(gca,'XLim',[0 255])
set(gca,'YLim',[0 10000])
title('Citra MSR Green')
subplot(3,3,6);
his(B(:),256,'FC','b','EC','b')
set(gca,'XLim',[0 255])
set(gca,'YLim',[0 10000])
title('Citra MSR Blue')

% --- button input citra asli
function pushbutton5
[nama_file,nama_path]=uigetfile({'*.jpg','File JPG (*.jpg)'},'Buka File Citra')
handles.citra = imread(fullfile(nama_path,nama_file))
guidata(hObject,handles)
axes(handles.axes1)
imshow(handles.citra)

% --- Histogram Citra Asli
function pushbutton4
Img = handles.citra;
RO = Img(:,:,1);
GO = Img(:,:,2);
BO = Img(:,:,3);

Merah = cat(3,RO,GO*0,BO*0);
Hijau = cat(3,RO*0,GO,BO*0);
Biru = cat(3,RO*0,GO*0,BO);

subplot(3,3,1);
his(RO(:),256,'FC','r','EC','r')
set(gca,'XLim',[0 255])
set(gca,'YLim',[0 10000])
title('Citra Ori Red')
subplot(3,3,2);
his(GO(:),256,'FC','g','EC','g')
set(gca,'XLim',[0 255])
set(gca,'YLim',[0 10000])
title('Citra Ori Green')
subplot(3,3,3);
his(BO(:),256,'FC','b','EC','b')
set(gca,'XLim',[0 255])
set(gca,'YLim',[0 10000])
title('Citra Ori Blue')

```

```

% --- Input button citra Median
function pushbutton7
[nama_file,nama_path]=uigetfile({'*.jpg','File JPG
(*.jpg)'},'Buka File Citra')
handles.citra3 = imread(fullfile(nama_path,nama_file))
guidata(hObject,handles)
axes(handles.axes3)
imshow(handles.citra3)

% --- Histogram Median
function pushbutton8
Img = handles.citra3;
Rm = Img(:,:,1);
Gm = Img(:,:,2);
Bm = Img(:,:,3);

Merah = cat(3,R,G*0,B*0);
Hijau = cat(3,R*0,G,B*0);
Biru = cat(3,R*0,G*0,B);







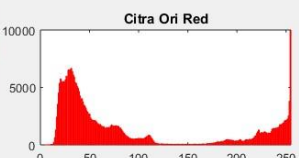
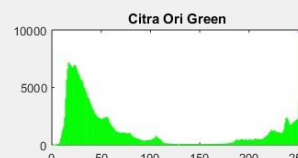
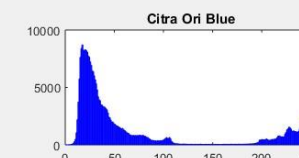
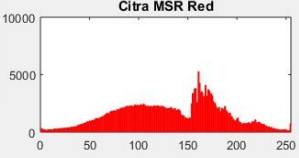
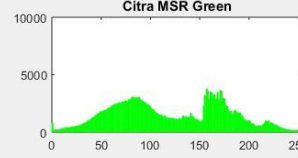
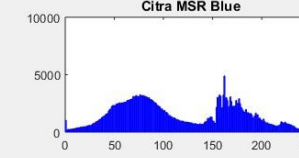
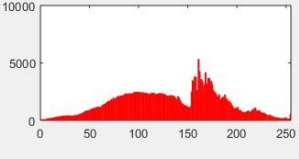
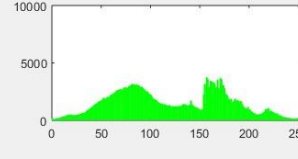
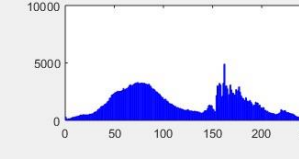



subplot(3,3,7);
his(R(:),256,'FC','r','EC','r')
set(gca,'XLim',[0 255])
set(gca,'YLim',[0 10000])
title('Citra Median Red')

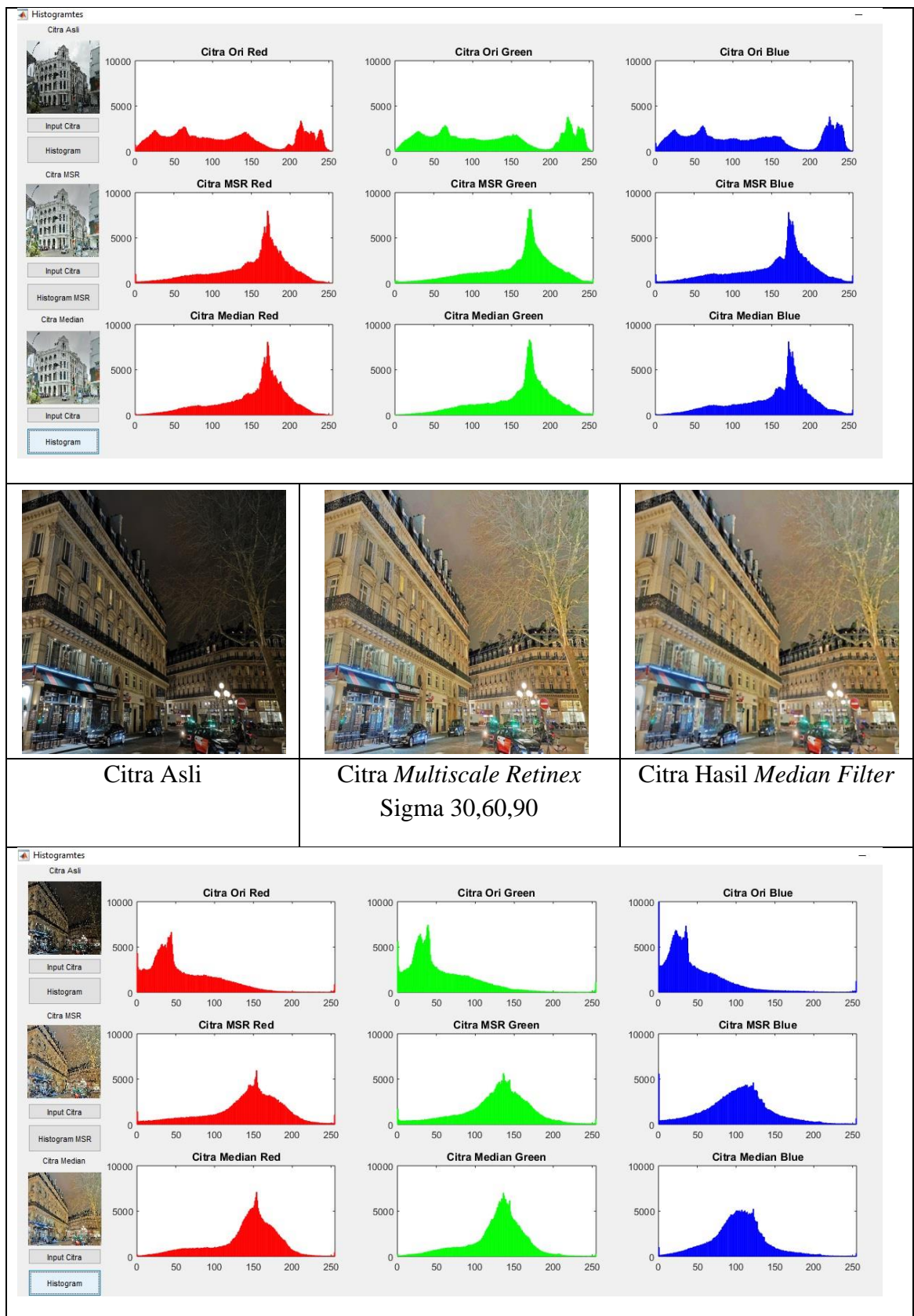
subplot(3,3,8);
his(G(:),256,'FC','g','EC','g')
set(gca,'XLim',[0 255])
set(gca,'YLim',[0 10000])
title('Citra Median Green')







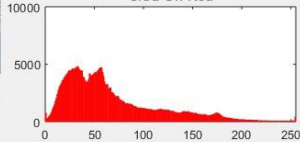
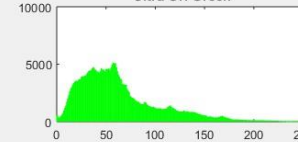
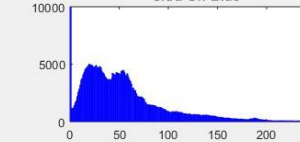
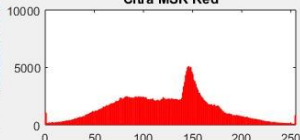
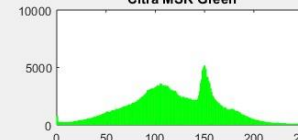
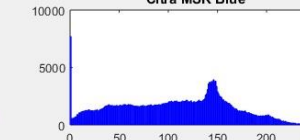
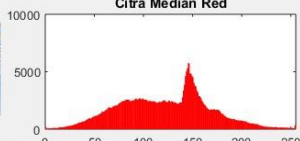
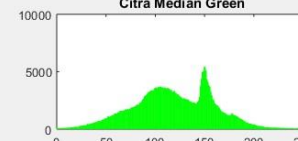
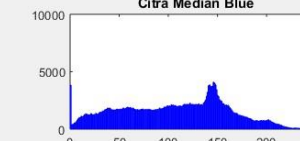



subplot(3,3,9);
his(B(:),256,'FC','b','EC','b')
set(gca,'XLim',[0 255])
set(gca,'YLim',[0 10000])
title('Citra Median Blue')

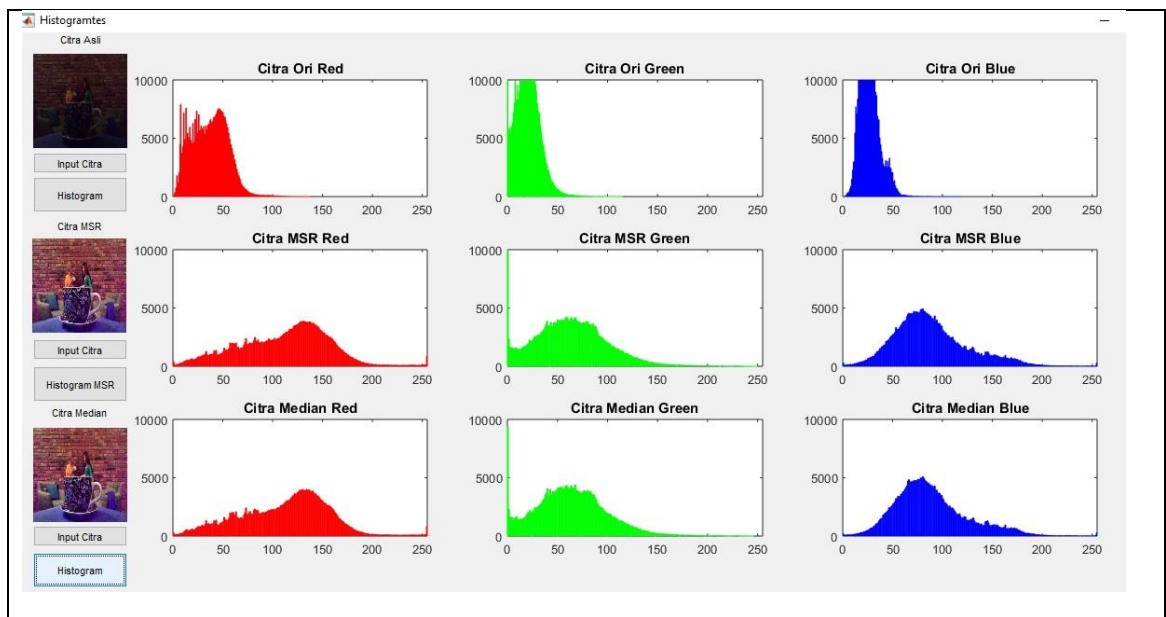
```

DATA UJI

Citra Asli	<i>Citra Multiscale Retinex</i>	<i>Citra Median Filter</i>
		
Citra Asli	<i>Citra Multiscale Retinex</i> Sigma 50,100,150	<i>Citra Hasil Median Filter</i>
<div> <div>  <p>Histogramtes Citra Asli</p> <p>Input Citra</p> <p>Histogram</p> </div> <div>  <p>Citra MSR</p> <p>Input Citra</p> <p>Histogram MSR</p> </div> <div>  <p>Citra Median</p> <p>Input Citra</p> <p>Histogram</p> </div> </div> <div> <div>  <p>Citra Ori Red</p> </div> <div>  <p>Citra Ori Green</p> </div> <div>  <p>Citra Ori Blue</p> </div> <div>  <p>Citra MSR Red</p> </div> <div>  <p>Citra MSR Green</p> </div> <div>  <p>Citra MSR Blue</p> </div> <div>  <p>Citra Median Red</p> </div> <div>  <p>Citra Median Green</p> </div> <div>  <p>Citra Median Blue</p> </div> </div>		
		
Citra Asli	<i>Citra Multiscale Retinex</i> Sigma 40,60,80	<i>Citra Hasil Median Filter</i>



		
Citra Asli	Citra <i>Multiscale Retinex</i> Sigma 20,40,60	Citra Hasil <i>Median Filter</i>
<div><div>Histogrames</div><div>Citra Asli</div><div><div></div><div><div>Input Citra</div><div>Histogram</div></div></div><div><div>Citra MSR</div><div></div><div><div>Input Citra</div><div>Histogram MSR</div></div></div><div><div>Citra Median</div><div></div><div><div>Input Citra</div><div>Histogram</div></div></div><div><div>Citra Ori Red</div></div><div><div>Citra Ori Green</div></div><div><div>Citra Ori Blue</div></div><div><div>Citra MSR Red</div></div><div><div>Citra MSR Green</div></div><div><div>Citra MSR Blue</div></div><div><div>Citra Median Red</div></div><div><div>Citra Median Green</div></div><div><div>Citra Median Blue</div></div></div>		
		
Citra Asli	Citra <i>Multiscale Retinex</i> Sigma 60,80,100	Citra Hasil <i>Median Filter</i>



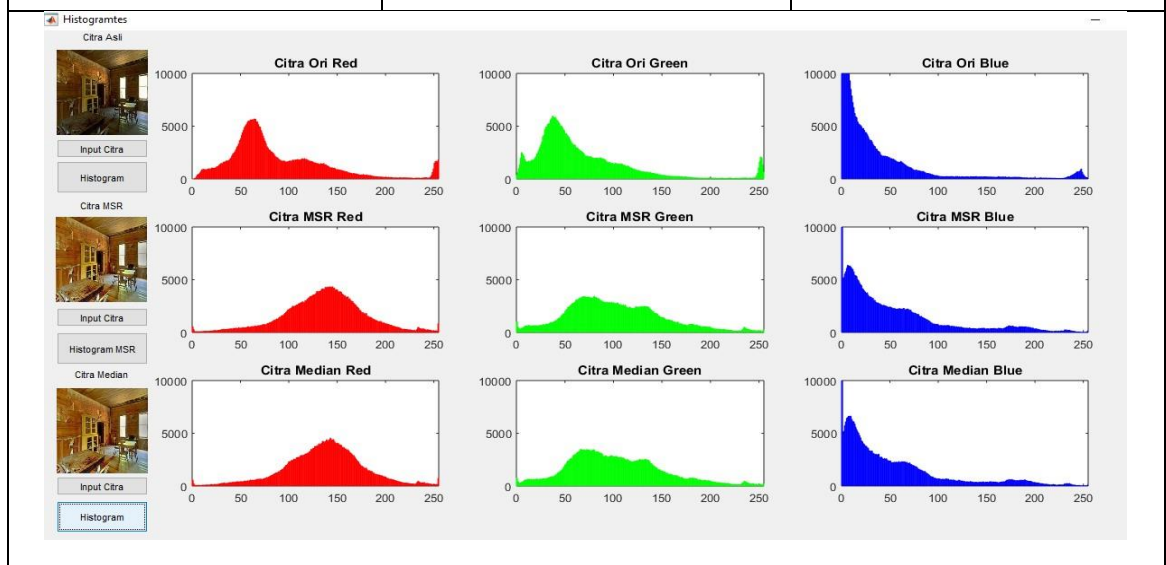
Citra Asli



Citra *Multiscale Retinex*
Sigma 30,50,70



Citra Hasil *Median Filter*

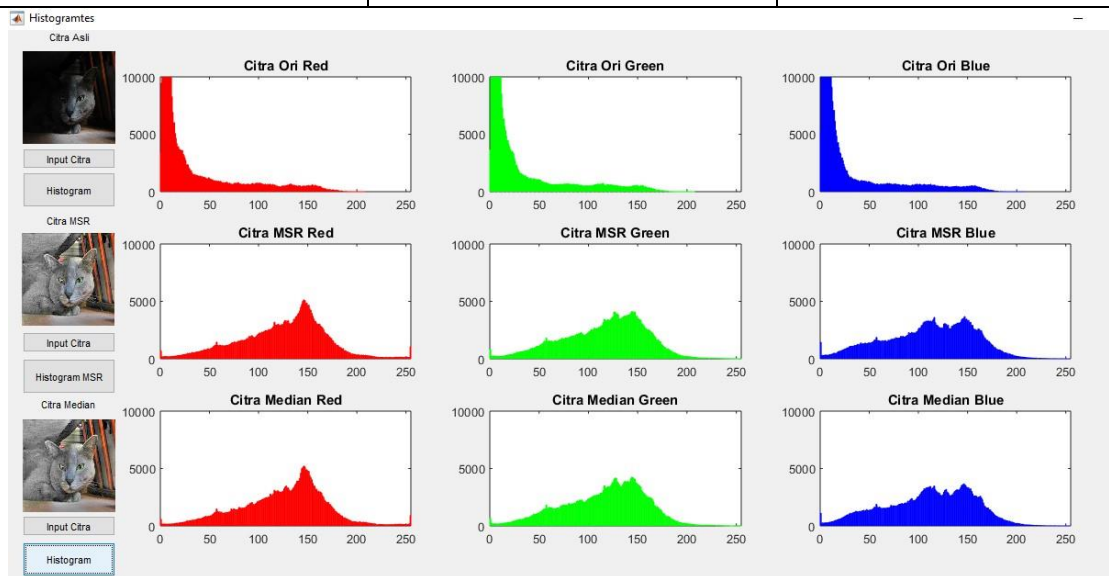




Citra Asli



Citra *Multiscale Retinex*
Sigma 20,40,60

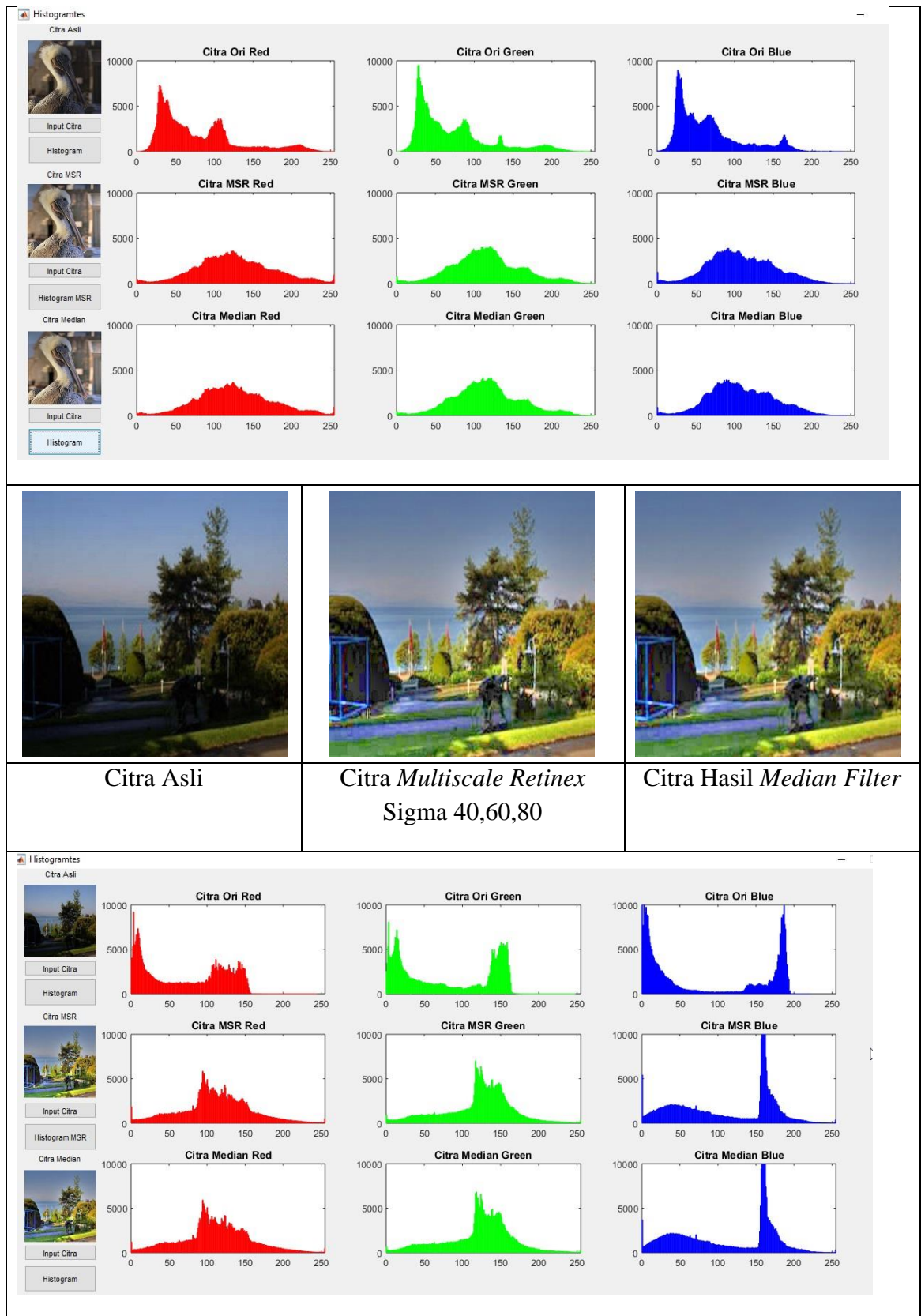
Citra Hasil *Median Filter*





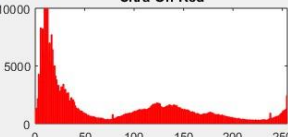
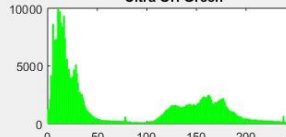
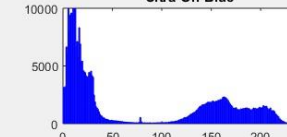

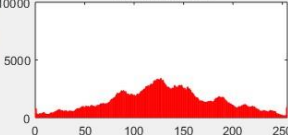
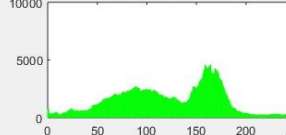
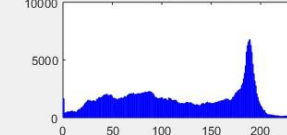

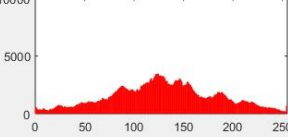
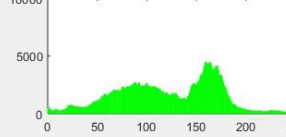
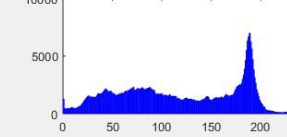



Citra Asli

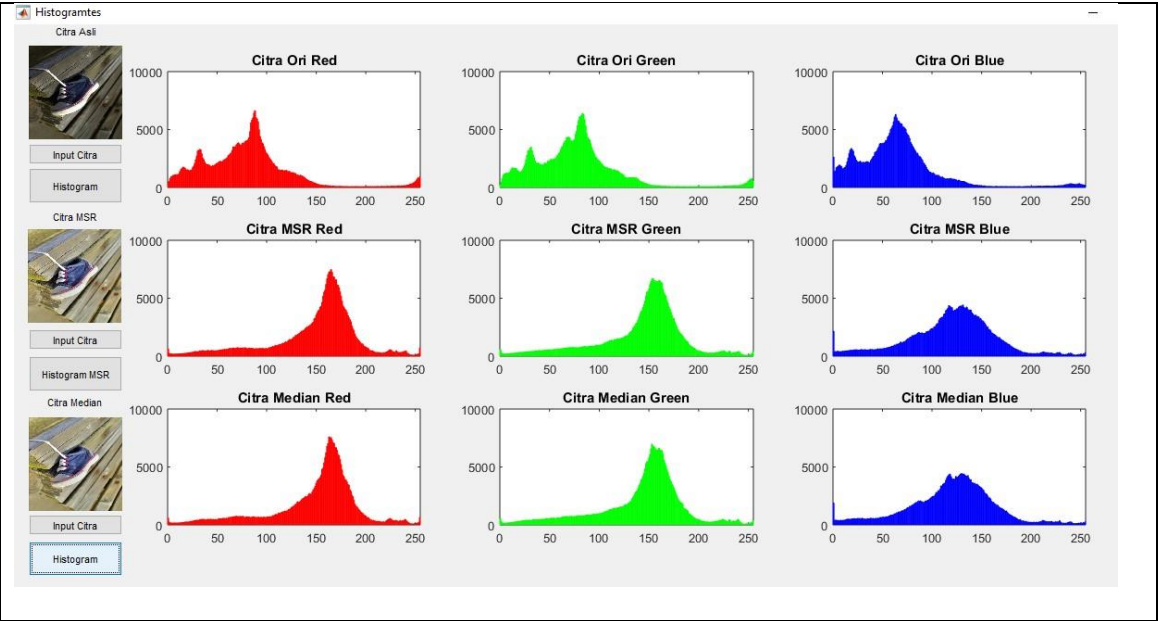


Citra *Multiscale Retinex*
Sigma 60,70,80

Citra Hasil *Median Filter*




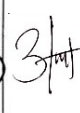

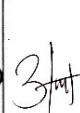

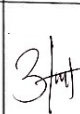

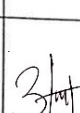
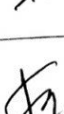
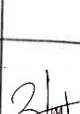
		
Citra Asli	Citra <i>Multiscale Retinex</i> Sigma 50,80,100	Citra Hasil <i>Median Filter</i>
<div data-bbox="300 824 1412 1400"> <div> <div>Histograms</div> <div>Citra Asli</div> <div>  <div>Input Citra</div> <div>Histogram</div> </div> <div>    </div> </div> <div> <div>Citra MSR</div> <div>  <div>Input Citra</div> <div>Histogram MSR</div> </div> <div>    </div> </div> <div> <div>Citra Median</div> <div>  <div>Input Citra</div> <div>Histogram</div> </div> <div>    </div> </div> </div>		
		
Citra Asli	Citra <i>Multiscale Retinex</i> Sigma 30,50,70	Citra Hasil <i>Median Filter</i>



KARTU BIMBINGAN SKRIPSI

Semester Gasal/Genap Tahun Akademik 2019 / 2020

Nama : RIZQI HIDAYAT TANUNGS	Pembimbing I : Dr. Mhd Furqan, S.Si., M. Comp. Sc
NIM : 0701163152	Pembimbing II : Abdul Halim Hasugian, M. Kom.
Prog. Studi : Ilmu Komputer	SK Pembimbing :
Judul Skripsi : Penerapan Kualitas Citra Digital menggunakan Metode Multiscale Retinex Dan Median Filter	

P E R T	PEMBIMBING I			PEMBIMBING II		
	Tgl.	Materi Bimbingan	Tanda Tangan	Tgl.	Materi Bimbingan	Tanda Tangan
I	14 Juli 2020	Revisi Bab I		30 Juni 2020	BAB I (latar Belakang)	
II	17 Juli 2020	Revisi Bab II & III		07 Juli 2020	BAB II (Tinjauan Pustaka)	
III	20 Juli 2020	Acc Proposal		10 Juli 2020	BAB III (Teknik Pengumpulan Data)	
IV	28 Juli 2020	Revisi proposal skripsi		11 Juli 2020	Penyusunan proposal skripsi	
V	28 Juli 2020	Revisi Bab III		12 Juli 2020	Acc proposal skripsi	

VI	20 okt 2020	Revisi Bab <u>IV</u>		31 Agus 2020	Revisi Proposal	
VII	21 okt 2020	Revisi Bab <u>V</u>		18 okt 2020	Revisi Bab <u>IV</u> & <u>V</u>	
VIII	22 okt 2020	Revisi Abstrak, kesimpulan, Daftar Pustaka		17 okt 2020	Revisi Abstrak, kesimpulan, Daftar Pustaka	
IX	23 okt 2020	Acc Semua Bab		20 okt 2020	Acc Semua Bab	
X	23 okt 2020	Acc Sidang		23 okt 2020	Acc Sidang	

Medan, 23 Oktober 2020

An. Dekan

Ketua Jurusan/Program Studi

Dr. Mhd Fergan, S.Si, M. Comp. Sc

NIP. 198008062006041003

Catatan: Pada saat bimbingan, kartu ini harus diisi dan ditandatangani oleh pembimbing

(DAFTAR RIWAYAT HIDUP)



Data Diri

Nama : Rizqi Hidayat Tanjung
Nim : 0701163152
Tempat & Tanggal Lahir : Pematang Setrak, 11 Mei 1997
Jenis Kelamin : Laki-laki
Alamat : Jln Bakaran Batu Lk II A
Kelurahan/Desa : Aek Kanopan
Kecamatan/Kabupaten : Kualuh Hulu / Labuhanbatu Utara
Agama : Islam
No Hp : 082267928501
Alamat Email : riskyntanjung214@gmail.com

Nama Orang Tua

Ayah : Drs. Marwan Tanjung
Ibu : Dra. Neni Triani

Pendidikan Formal

-SD : SDN 1144339 Kuala Bangka
-MTS/SMP : MTS Muhammadiyah 16 Sei Rampah
-SMK/SMA : SMK Negeri 01 Teluk Mengkudu